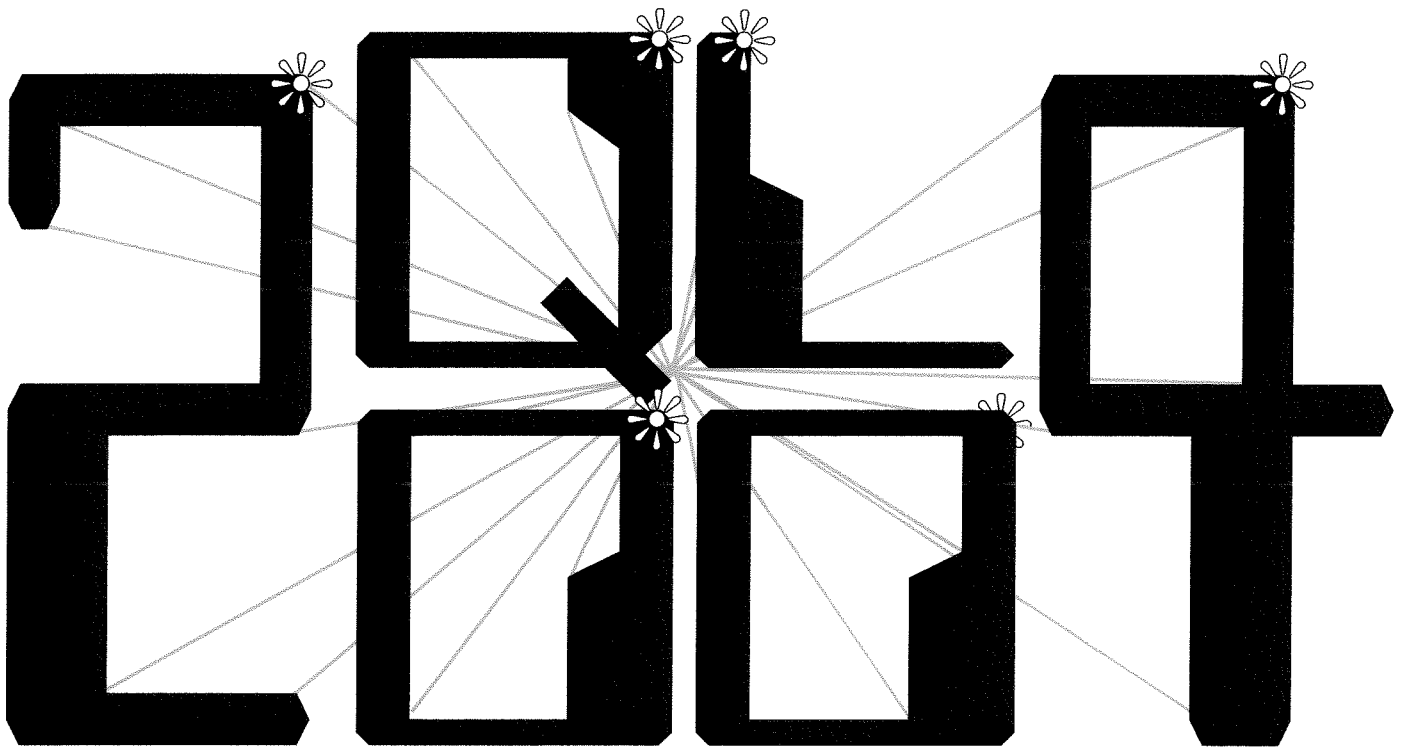


# QL Today

Volume 8  
Issue 5  
Jan./February  
2004

ISSN 1432-5454

The Magazine about QL, QDOS,  
Sinclair Computers, SMSQ...



Another Main QL Event coming  
up? Plans for QL2004 inside!

## COLOURS

with  
EasyMenu

Wolfgang Uhlig  
explains  
how to do it

## QLTdis

### Next Part

Norman Dunbar's  
Disassembler Project  
is getting closer

# Contents

3	Editorial	
4	News	
8	Colours with EasyMenu (with special regard to the new system Colours)	Wolfgang Uhlig
11	Turbo and Parameters	George Gwilt
16	Programming QPTR in BASIC - third part	Wolfgang Lenerz
20	QLTdis part 11	Norman Dunbar
31	Corrections	
34	Windows in the Pointer Environment using SETW	George Gwilt
36	Letter-Box	
37	Tiptoeing Through Turbo	George Gwilt
38	The new GD2 sprite system	Marcel Kilgus
42	Launchpad Update	John Perry
48	Hints and Tips for QPC	Jochen Merz
49	London Quanta Workshop, Jan. 2004	Roy Wood
50	Byts of Wood	Roy Wood
55	QL 2004	Geoff Wicks

## Advertisers

in alphabetical order

D & D Systems	13
Dilwyn Jones	53
Jochen Merz Software	19
Phoebus R. Dokos	
- Quantum Leap Software.	47
Q-Celt Computing	41
QBranch	32, 33
Quanta	43
RWAP	23
TF Services	39
Geoff Wicks	7

## QL Today

ISSN 1432-5454

### German office & Publisher:

Jochen Merz Software    Tel. +49 203 502011  
Im stillen Winkel 12    Fax +49 203 502012  
47169 Duisburg        BBS +49 203 502014  
Germany                email: JMerz@j-m-s.com  
                              email: QLToday@j-m-s.com

### English office:

Q Branch                Tel. +44 1273 386030  
20 Locks Hill            Mobile +44 7836 745501  
Portslade                Fax +44 1273 381577  
BN41 2LB                email: qbranch@qbranch.demon.co.uk  
United Kingdom        email: QLToday@j-m-s.com

### Editor:

Dilwyn Jones            email: dilwyn.jones@tesco.net  
41 Bro Emrys            email: QLToday@J-M-S.com  
Tal-Y-Bont, Bangor  
Gwynedd  
United Kingdom LL57 3YT

### Co-Editor:

Bruce Nicholls         Tel +44 1708 510764  
38 Derham Gardens     Fax +44 870 0568755  
Upminster                email: qltoday@q-v-d.demon.co.uk  
Essex RM14 3HA        email: QLToday@j-m-s.com  
United Kingdom

**QL Today** is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in \_SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April  
Issue 2: 30 June  
Issue 3: 30 August  
Issue 4: 30 October  
Issue 5: 30 December  
Issue 6: 28 February

**QL Today** reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 2003 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

Here we are in 2004, 20 years after the launch of the QL, as Urs König recently reminded us via the QL mailing list:

"On January 12th 1984 (exactly 12 days earlier than Steve Jobs lauched the Apple Macintosh) Sir Clive Sinclair presented the QL professional computer to the world in a hollywood-style event at the Intercontinental Hotel, Hyde Park Corner, London."

20 years on and we have such QLing delights as colour drivers, Q60s, various QL emulators, net access, graphical user interfaces and so on, many things we could only have dreamed of years ago. Just what is it about QDOS and SMSQ that keeps us going after 20 years!

One of the latest developments is the new Window Manager and software designed to ease the use of the colours, such as QCoco by Wolfgang Uhlig. Wolfgang has an interesting article about getting hi-colours in Easymenu. All this contributes towards the maturity and usability of the new colours.

It looks like there will be a busy QL show season this year. Plans have been made for a large QL2004 show later in the year and there will be several other shows during the year as well. The USA QL show now has its own website – see elsewhere in this issue.

There are changes afoot at Quanta. Unfortunately, the new editor got off to an unfortunate start, with problems delaying his first issue, resulting in the eventual production of only a single issue to span 3 or months, and to top that the Chairman and Secretary are to step down this year after several years of much valued contribution. It is to be hoped that Quanta will be able to appoint enthusiastic new officials to carry the group forward to the future.

Happy QLing to you all in 2004!



Cartoon

# NEWS

## QCoCo

### *Wolfgang Uhlig writes:*

QCoCo is now version 1.5. It has integrated the new bmpcvt-extensions by W. Lernerz which make background images look fine on the Q40/60, too. It is compiled and has a config block level 2, which makes configuration much easier. The RGB colour-mixer is integrated now, more compact and convenient. If you use QCoCo, it must be said that despite the possibility to apply lots of 3D-borders you can't take any border anywhere! The Qpac2- and QMenu-menus are of fixed sizes and if you give them borders that are not QL-compatible, you might experience strange behaviour. Please avoid therefore to give an Application Window a 3D-border from 1030 up to 1039. This is the entire last row and the first and last button of the first row in the border-table in QCoCo.

I shall redesign this table to make it clearer, when I have the time. Also I heard of people having strange crashes or half drawn windows. This has to do with older WMAN-stuff that is loaded in the bootfile. Please look that only SMSQ/E 3.04 is used and no other WMAN-extensions. Another user told me of a possible problem with the older version of bmpcvt-extension of W. Lernerz. There is a new version in order to make bitmaps look fine on a Q40/60, too, and QCoCo has this already integrated. I cannot reproduce this problem on my machine, because both versions run together without problems, but you could check this, if there are problems. I am aware of the fact that the older version is needed to run Qcolour, the RGB-mixer. But first, as I said, on my machine both run fine and second, there will be a new RGB-picker in a short time, I have already started to design it. Again, Marcel provided me with an extension which will make that possible. QCoCo can be downloaded at: [www.uhlich.nl/ql/qcoco.zip](http://www.uhlich.nl/ql/qcoco.zip)

## Editor News

### *Mark Knight*

Mark has been working on Editor again, prompted partly by George Gwilt. George informed him that Editor's method of finding out how much free memory was present was not compatible with latest versions of SMSQ/E running on some hardware with stacks of RAM, so it needed a compatibility update. Editor did still work on these

systems but couldn't use all the memory present. George also provided a sequence which could be typed into Editor to reliably reproduce the one known bug left in it, plus some information that tied it down to an initialisation bug. Once Mark had the ability to reproduce the bug and the extra information it actually took less than half an hour to pin it down and fix it.

In order that the new version can be used the latest Turbo Toolkit is needed, this is in the .ZIP file on Mark's web site so you get it when you download, just do remember to install it. Those who want to link the toolkit into the executable, Q-Liberator style, will need the source code and the latest Turbo compiler, available elsewhere. The .ZIP files need to be ported across to your QL or SMSQ/E system and unzipped there, or get the disks from a QL public domain trader once they have downloaded the files. Mark's web site is at: <http://www.the-furnace.demon.co.uk>

...and the Editor 3.07K is on the "QL Stuff" page. So that it is easy to distinguish the program from previous versions it announces itself as "Editor 2004" on the start-up screen. By all means visit other pages on the web site, particularly the art galleries if you like pencil drawing or computer graphics artwork.

## USA QL Show 2004 Website

### *Jim Hunkins writes:*

The web site for the US QL Show 2004 is now on line at:

<http://www.jdh-stech.com/ql-usashow.htm>

If you are thinking or planning on attending, please let us know as soon as possible and also get your reservations in as I understand that Florida is fairly busy that time of year.

I have included a fairly extensive travel page with all kinds of links to transportation, motel reservations, and area things to do for your convenience. You might also note that the email links will all require Java 1.1 or later (anti-spam encoded) but you can also pick up the addresses from the graphics on the contacts page.

Enjoy and looking forward to seeing you all there! Thanks to Phoebus Dokos, we have added several new items to the QL USA Show schedule on the web, along with some additional updates. We are now showing that we will have 6 exhibitors, four from Europe. Please take a look at the site and let me know of any changes, if you plan to attend (and if we can list you), if you plan to exhibit and aren't yet listed, and any ideas or volunteering for more demos/talks, etc.

Also, watching the recent thread for the QL 2004

(on ql-users email mailing list), I have added tentatively some kind of panel discussion. I do recall that they have always been popular at previous shows.

## BMP2SPRT

Wolfgang Lenerz has written a small QL program to convert PC bitmaps into QL sprites.

You can find v1.10 of BMP2SPRT on Wolfgang Lenerz's website:

[www.scp-paulet-lenerz.com/14mljkl24/Wolf/download](http://www.scp-paulet-lenerz.com/14mljkl24/Wolf/download)

Also, please note Wolfgang's new email address [wolf@scp-paulet-lenerz.com](mailto:wolf@scp-paulet-lenerz.com)

## SMSQ/E Sources Website

Please note that the SMSQ/E sources may be obtained at the following website:

[www.scp-paulet-lenerz.com/smsqe](http://www.scp-paulet-lenerz.com/smsqe)

The site is still under construction, so please be patient. It is foreseen to include the following for download:

individual directories within the sources "upgrade" package from one version to another (but only from version x.xx to version x.xx + 1)

## IFE Library News

### *Javier Guerra writes:*

Thanks to the work of Miguel Angel Hernandez and Salvador Merino, it's available now, on the internet, the discs of the IFE library at:

<http://homepage.mac.com/miguelahernandez1/Education14.html>

and

<http://homepage.mac.com/WebObjects/FileSharing.woa/1/wo/ahfv6UZclRblfeRS.1/0.2.1.2.22.31.97.1.35.0.1.1.1?user=miguelahernandez1&fpath=IFE%20Library&templatefn=FileSharing1.html>

The Independent Spanish Users Club of Sinclair QL (CEIUQL) (1985-1997), known also as QLAVE (who sounds as "clave" in Spanish, and means "key" or "response"), was a member of IFE (Inter-group Freeware Exchange), a kind of Federation of International Clubs whose only objective was to create a great library of free QL software.

## Q-Emulator News

### *by Daniele Terdina*

Just a quick note that Q-emuLator 2.3.1 is available at <http://users.infoconex.com/daniele/winql.html>

This is a minor update adding more TCP/IP functionality (needed for email and FTP) and a fix for a bug affecting the PAR device on Windows XPAs always, the update is available for free to all registered users.

## Text87 News

### *Jochen Merz*

text87, one of the few programs which does not run properly in high-colour mode can now be modified to work in 16-bit colours on the QXL and QPC (with SMSQ/E Version 3). Marcel Kilgus has added a new driver for this program which will be added to the program and enable you to run it nicely in high-colour mode. You have two options: it can look the same way as before (white/red/green on black) or - much better looking - colours on a nice grey. It is up to you, you can try both settings.

text87plus4 PATCH is available now from J-M-S for only EUR 12,- including postage & packaging! Easy-to-use, will it provide you with a text87 which works without having to re-start QPC in a different colour mode or use a DISP\_COLOUR 0 which will screw up the rest of the display. No, with text87plus4 PATCH it will work without the need to change display resolution - and we think it looks even better in grey!

Please note: the PATCH program requires you to own the latest version of text87plus4 English (E4), file size 116850 bytes or the latest version of text87plus4 German (g4), file size 117354.

You do not need to configure or modify the result of using text87plus4 PATCH-just execute this one instead of your original program!

Please note: the Patch is sold with the permission of text87 author, Fred Toussi. Currently, it only works on QXL and QPC in high-colour mode (not 8 bit) and not on GC and Qx0 either. Depending on the interest, Marcel may consider implementing more screen drivers and modes.

You can now find a secure contact form on the J-M-S homepage ([smsq.j-m-s.com](http://smsq.j-m-s.com)), where you can submit credit card data etc. without having to send it via email.

## News from Marcel Kilgus

Just a quick note that I have updated my pages at [www.kilgus.net](http://www.kilgus.net) with a new "SMSQ/E" section that includes various new stuff that is not directly related to QPC, like my WMAN theme, various updated applications and a Windows program that converts PNG images to SMSQ/E sprites (most new sprites in SMSQ/E and the JMS collection have been converted using this program after being created with Photoshop as it also handles the alpha channel correctly).

## CYBIKO PDA Driver for QDOS

Simon has written a QDOS driver for a Cybiko. A

Cybiko is a nifty cheap Russian PDA – one of the few around that still uses a RS232 serial cable to link to a PC/Mac (and now a QL, thanks to Simon) instead of USB. They are available in 3 colours, and cost just 20 pounds sterling each – dirt cheap, considering what you get.

see [www.cybiko.com](http://www.cybiko.com)

They can email and internet when connected to a PC/Mac, they can text message other cybikos for free, and they can download and install applications, including a word processor – an application CD costs just five pounds!

## QL Documentation Online

by *Dilwyn Jones*

I have now put much of the QL Documentation CD online on my TopCities website. Either go to the normal Tesco.Net site

<http://homepages.tesco.net/dilwyn.jones/index.html>

and click on the new link at the bottom of the page or go direct to the site on

<http://dilwynjones.topcities.com/index.htm>

and click on the QL Manuals & Documentation page link. There's all sorts of useful articles, replacement manuals and QL-related documents there.

Not all of the material is there yet, but it's a huge improvement on what I was able to put onto the Tesco.net site because of space limitations.

Anyone with additional material to put online is welcome to send it to me for inclusion. I hope this will prove to be a valuable online resource of QL related documentation and information.

This site also hosts a large amount of old QL-zines (back issues of Club QL International newsletters, QL-Forum, QL Hackers & Linux Journals, Italian magazines and in due course a pointer to the Spanish QL Resources site to access the Spanish QL newsletters too), as well as being the new home of the QL Rogues Page, Pictures of QL related equipment you can use for creating your own websites, the QL Humour page (including the complete collection of QL Today cartoons) and the back issues of the QL Today cover disks.

## LAUNCHPAD V0.96

Launchpad has now been updated to v0.96. Below is a list of new and updated features in this version (the first paragraph refers to changes in version 0.95 – the version number is shown in the Copyright menu or while you configure the program).

Launchpad now only does an RJOB "CLOCK" if the Launchpad clock is active, in order to prevent the unintentional removal of other jobs with the

name 'clock' such as the QPAC1 clock program. The MySQL menu now uses EX to start Q-Trans rather than EW, to allow you to CTRL C between Q-Trans and Launchpad if required. Desktop menu names now default to Menu 1, Menu 2, Menu 3 and Menu 4.

Added Jobs menu, identical to Utility-Pick/Remove A Program, but quicker to access. Made sense since launching a program is so quick and easy, but Pick or Remove a job is a bit more long winded in older versions.

Updates from earlier versions are free of charge (see previous issue's news page for details).

For Qlers in Germany, Austria or Switzerland, Launchpad may now be ordered via Jochen Merz Software if preferred. Customers wishing to pay in Euros should preferably order from Q-Celt Computing outside those areas.

## QL Today Index

*Timothy Swenson* has produced a QL Today Index covering the volumes after volume 4 (my original index only covers the first few volumes). The index is currently in Excel v5.0 format, though it should be possible to export this into text or CSV format, for example. By the time you read this, I should have been able to add this to my website and PD library.

## PD Library News

*Dilwyn Jones*

My PD library now consists of 167 disk of general QL PD software. Recent changes are as follows:

### disk GE163:

Mark Knight Utilities disk of programs, snippets and useful toolkit-style extension files. The programs include: Auto Indent Program / BList, SuperBASIC Listing program / CONFIGURE / Condense, the SuperBASIC/SBASIC file condenser / File Manager version / Printer Install Program / QL character set editor, QuickSet / QL Super Alarm program

### disk GE164:

Floppy Disk Utilites v1.22 from Ergon Development. Especially designed for the (Super) Gold Card user. This program supports DD, HD (1.44 MB) and ED disks (3.2 MB). Disk editor, single/multiple disk copier / formatter / verifier. It works also with a single disk drive. File recover, Search, Collect, Print sectors. You can also format disks with an extra 5% capacity (a bonus of 160KB with ED disks). If you have a 3.2MB or 1.44MB disk drive this program could be very

useful (SMS2+QVME compatible). The manual is included (Italian and English language) in Quill format. PopCalc - a popup calculator program from Mark Knight. SetDate - set the QL clock effortlessly with this utility. Shrink - Memory Shrink v2.50, reset a QL to a given memory size. Super Kit Merger - merge your extensions as one toolkit. Soft Eprom v1.00 - Save ROM images to file to be run from RAM. Tools v1.07 - a head-up display of date, time, caps lock, etc. Touchit - programmable keys v2.15

**disk GE165:**

TextTools, supplement to GNU Text Utilities - see disks GE118/9. This package is version 1.1, ported by Franz Krojer.

**disk GE166:**

(4 disks) PGP v2.6i - Pretty Good Privacy encryption software, ported for QL by Jonathan Hudson.

**GE167:**

Recover - Archive database recovery program. BASIC version included. ResQL - Hans Lub's program for recovering damaged files or floppy disks. Sector Editor - Sector editor for DD floppy disks, by Graham Underwood. Unzip Librarian -

Pointer driven and non-pointer driven versions of a front end program for QL unzip. Helps make use of Unzip that little bit easier! Universal Turing Machine simulator, by Kevin Betts. Utilities - a set of short programs by Per Erik Forssen. Includes Clock v1.26, ESCstrip v1.04, Disk Re-labeller v1.00, OpenFile v0.05, Snapshot v0.01.

**EPSON Printer Codes**

*Dilwyn Jones*

I have prepared a reference document of Epson printer control codes, using the standard Epson ESC/P 2 documentation supplied on my QL Documentation CD. The snag with the Epson PDF files is the sheer size, several megabytes and you need the Adobe PDF file reader programs to read them. My document is a cut-down document, in Quill format. It will be short enough to email or place on my website when complete (nearly complete at the time of writing) alongside the existing HP PCL printer documentation.

Please send any QL-related news you would like to see published here to  
[Editor@qltoday.com](mailto:Editor@qltoday.com)



**JUST WORDS!**

## The final countdown...

**QTYP dictionaries:**

*Danish - 23,515 words*  
*Dutch - 180,130 words*  
*French - 208,913 words*  
*German - 165,935 words*  
*Italian - 83,829 words*  
*Norwegian - 61,413 words*  
*Spanish - 174,846 words*  
*Swiss German - 165,810 words*  
*UK English - 82,098 words*  
*USA English - 77,722 words*

£1 or €1.5 EACH

**Programs:**

*QL-2-PC TRANSFER*  
*QL-RHYMES*  
*AUTO-GRAPH*

£10 or €15 EACH

**Plus:**

*VOCABULARY DATABASE*

£5 OR €7.50

*Geoff Wicks, 56 Peveril Crescent, West Hallam, Derbyshire DE7 6ND, U.K.*

*Tel: +44 (0)115 - 930 3713*

*email: [gwicks@beeb.net](mailto:gwicks@beeb.net)*

*Web: <http://members.lycos.co.uk/geoffwicks/justwords.htm>*

# Colours with EasyMenu

(with special regard to the new system Colours)

Wolfgang Uhlig

In the last QL-Today Geoff Wicks wrote some comments on a review of one of his programs. One of the statements was that EasyPtr is unable to use the new colour drivers. Have a look at my programs QColour, MfColour and QCoCo and you will know that this is not true. The truth is: you cannot use the new colours when designing menus inside EasyMenu (and even this is no longer completely true, because you can use the new colour sprites, see below), but you can make EasyPtr programs use the colours without nearly any limit! And that's what this article is about. It's written for people who already know how to design menus in EasyMenu, to append menus and sprites to an appendix-file and to write a slightly advanced EasyPtr-program. I recommend that beginners first read the articles about EasyPtr-programming I wrote in former editions of QL-Today. Also I want to give some explanations about the new system colours from the point of view of somebody who has no assembler programming skills at all and only little understanding of bits, bytes, hexcodes etc. I think that there are more people among the QL-community like me and I want them to profit from what I have learnt while I was programming QCoCo, the new system-colour-configurator. One of my essential "knowledge sources" was Marcel Kilgus to whom I am deeply indebted (to take Geoff's beautiful expression).

The trick is actually very simple: EasyPtr allows you to make any window the active window and accept all graphic commands like *PAPER*, *INK*, *BLOCK*, *BORDER* etc. by the command: *MWINDOW*. So you may also use the new system-colour commands *WM\_PAPER*, *WM\_INK*, *WM\_BORDER*, *WM\_BLOCK* etc. The most convenient windows to handle are *Information Windows* (from now on: *lFW*) and *Application Windows* (*AppW*). But also *Loose Items* (*LI*) can get colours but have to be handled a little bit differently.

*lFWs* (which can even overlap each other) are not checked in a *MCALL*-routine and keep their colour information throughout the whole program loop. It's enough to give them a colour in the very begin of the program. So the command  
`MWINDOW #3,1! : WM_PAPER #3,32799 : CLS #3`

will give the first *lFW* in your program (which has opened a *con\_* channel 3) the colour blue and this will remain as long as you don't give it another colour or close the program. If you have overlapping *lFWs*, you have to define the windows in the correct sequence, the last defined will always be on the top. The only drawback is that you cannot use the *Info Objects* any longer because they disappear when giving the *lFW* a colour. Here the *AppWs* are the solution as you can write text into them. Place as many *AppWs* into a *lFW* as you originally wanted to place *Info Objects*. Design them without border. You want to have a blue background and four small texts in white ink on it for example? Then write following code in the program:

```
MWINDOW #3,1! : WM_PAPER #3,32799 : CLS #3 FOR
i%=1 to 4
  SElect on i%
    =1:a$="text one"
    =2:a$="text two"
    =3:a$="text three"
    =4:a$="text four"
  END SElect
MWINDOW #3,i% : WM_PAPER #3,32799 : WM_INK
#3,65535 : CLS #3 : PRINT #3,a$
END FOR i%
```

You want to have a 3D-border around your *lFW*? No problem!

```
MWINDOW #3,1! : WM_PAPER #3,32799 : WM_BORDER
#3,2,1026 : CLS #3
```

will do the trick (the colour values are explained later in this article). And of course the *AppWs*, too, can have a border in the same way, you have only to take care that the border-width (in the above example=2) leaves enough space for the text. If not, you have to enlarge the *AppWs*. *AppWs* cannot overlap each other and they are checked in the *MCALL*-routine. You have to take care what happens to them during the program loop.

*LIs* are special because they change their status and are therefore redrawn often. Every time, a *LI* is clicked on, you will have to apply the colour anew AND to set the status in order to redraw the button. See the following code:

```
MWINDOW #3,-4
WM_PAPER #3,32799 : WM_INK #3,65535 :
WM_BORDER #3,1,1026 : CLS #3 : Print #3,"I am a
button"
stat%=MSTAT%(#3,-4 to 0)
```

This gives the *LI* number 4 a blue background, white ink, a 3D-border of 1px width, the text "I am a button" and makes it appear by setting the



status. There are different ways to manage this. If you have many LIs, you might prefer to write a procedure or function to control this. You may even give different appearances to the same button, dependant on its status. Write a procedure that checks the status of the LIs and gives them different colours and/or borders. Adequate commands are the above mentioned function MSTAT% or the procedure MSTAT. See the EasyPtr manual for further explanation.

Also, if you have to apply colours very often, you could help yourself by defining all colours you will use, at the begin of the program

```
grey=60250
pink=64543
red=64512
etc.....
```

and then write procedures like:

```
DEFine PROCedure P (chnr,col)
  WM_PAPER #chnr,col
END DEFine P
```

for PAPER (and for INK and BORDER similar), which you can use with P3,pink. It will save you a lot of typing.

There is **one bug in EasyPtr** which I had never noticed in "colourless" times: Never try the MWINDOW command on LI number one. It will not work! Don't ask me why.

The only window that cannot have colour is the **main window**. Here, too, the trick is simple. Make your very first lfw as large as the main window, give it a colour and you are there!

A very special and joyful thing is the use of **pictures as a background** or just in a window. Wolfgang Lenerz has provided us with an extension that will do the job. Together with his new program BMP2SPR comes the bmpcvt-extension. In this extension there is one command which is very interesting for EasyPtr-programmers. It's called WL\_bmp8load and loads an 8bit-bitmap into a channel. This can be any normal SBasic-window and thus an Easymen-window, too. The Easyptr-command that will do it, is: MWLINK. With this command you link a channel into a window. For example:

```
lnkchannel1%=FOPEN(con)
MWLINK #3,1!#lnkchannel1%
WL_BMP8LOAD #lnkchannel1%,"win1_my_picture.bmp"
```

opens a new free channel, links it to lfw 1 and loads "my\_picture.bmp" into it. Simple, isn't it? The

only important thing is, that it is really a **256 colour** bitmap. Even the size is not very important, if it is too large, it is clipped!

The newest development "on the market" are programs with which you can create **colour sprites** that have the format you need for EasyPtr. There are (afaik) two programs to produce them, one is the above mentioned **BMP2SPR** by W. Lenerz and another one is a small PC-utility, **PNGCVT** by Marcel Kilgus, which transforms PNG-files to QL-sprites. You can use them in Easymenu and the good news is: you can even see them in Easymenu itself! The bad (but not really bad) thing is that you can only take sprites of a maximum size of 64x64 pixel in Easymenu and when appending them to an EasyPtr-appendix file. You may take sprites much larger than that, but you will have to store them anywhere and load them into your Basic-program during runtime.

There is one new feature in BMP2SPR which might turn out to be one of the most beautiful things: You can make an 'extended' sprite, which means that such a sprite can have up to 5 different appearances dependant on its status. I didn't have time to profit from this feature in QCoCo because of its being brandnew but I already know that it works like any 'normal' sprite and that is fantastic! You should get your copy of BMP2SPR and read the manual for further explanation.

There are two ways of using a sprite in an EasyPtr-program. The first one is to add it to a LI as an object as described above. The second one is to apply an appended sprite with either the SPRS or SPRW-command within the program. For example, with

```
SPRS #3,1
```

you make the first sprite in your extension file the pointer for your program. With

```
MWINDOW #3,4!:SPRW #3,0,0,2
```

you make the **second** sprite in your extension file appear in lfw 4. As I said in my first EasyPtr tutorial, the manual isn't quite clear at this point, because 'name' or 'adr' must actually be the number of the sprite in the appendix-file, where it is automatically placed after all the menus.

There was one odd thing I noticed when working with this: When you give LIs a sprite (*Loose Item-properties-object* in Easymenu) you will see them during your Easymenu-session. You can save the menu apart or attach it to an appendix-file and when running a BASIC-program with this menu, everything is just fine. However, when you load this menu into Easymenu again, the sprites are gone and you have to load them

anew. I recommend to put them into the menu in the very end of menu development. Actually, the planning of an Easymenu-menu is more important now than it was. You will probably have many more windows to hold all your information and in order to have good control it is very practical to give them different colours in Easymenu. You still have lots of stipple colours to manage that.

Okay, now a few more words about the system colours. As you could see, I took the decimal values in the commands described above. I want to tell you why. A lot has been written about the system colours in past QLToday issues. Mostly it was information from assembler programmers for other assembler programmers. I cannot judge for other people but as to me, I was frustrated because I didn't understand much of it. When programming QCoCo I had to learn what it's all about and with the help of Marcel I succeeded to a certain extent. Here's what I can tell those who have the same problem I had:

When it all started we talked about 24bit colours, also called 'true colour'. But in reality the GDI-driver can handle a 16bit range of colours which is 'only' 65535. So, if you write your SBASIC-program in mode 'colour\_24' you give HEX-values like \$FFFFFF (white) for example but the system will show you the system colour 65535 (or \$FFFF) which of course is white, too. In colour\_24 mode you even HAVE to give Hex-values because the decimal values are not interpreted correctly.

It is therefore much easier to forget about the 24bit colours at the moment and take only the system colours by using the 'WM\_'-commands (see above) together with a Hex-value or decimal value, whatever you prefer. The big advantage is, that your program will be 'understood' by any machine running in 16bit colour mode and if you give special windows 'special' colours, they will even show up in the colours the user of your program has defined on his machine.

## How does it work?

In the whole range of 65535 colours there are special ranges for special colours:

- 1) From 0 to 255  
(Hex \$0 to \$FF) are the normal QL-colours
- 2) From 256 to 511  
(Hex \$100 to \$1FF) are the palette colours (COLOUR\_PAL)
- 3) From 512 to 767  
(Hex \$200 to \$2FF) are the **systemcolours (references)**

- 4) From 768 to 1023  
(Hex \$300 to \$3FF) are grey 'colours'
- 5) From 1024 to 1279  
(Hex \$400 to \$4FF) are the **3D-borders**
- 6) From 16384 to 32767  
(Hex \$4000 to \$7FFF) are all stipple colours
- 7) From 32768 to 65535  
(Hex \$8000 to \$FFFF) you have the **RGB-colours**

Nearly all these colours you can use by only typing `WM_PAPER 17484` or `WM_INK $8F8F` (Hex or decimal is your choice). Two ranges, however, are special: range 3) and 5):

Range 5) are the 3D-borders and its obvious that you must not write `WM_PAPER` or `WM_INK`, but only `WM_BORDER` with these values. Also there are not so many 3D-borders that really work well. Marcel had defined some good looking borders which were shown in a former issue of QLToday. You can also find a picture of them together with their values in my program QCoCo. What you can try are 3D-borders from 1024 to 1039 (\$400 to \$40F). Other ones may give strange results. The most important range for the SBASIC-programmer is range 3). The first 57 of these are **references to the system palette colours**. These are the colours that are used in all QPac2-menus, QD, FiFi, QSpread etc. They have strange names like 'Application window item unavailable foreground' or cryptic ones like 'Subsidiary information window middleground'. Don't mind, it's much less cryptic than it looks. The tool to see what they are and do is again QCoCo. Once you have understood them you will want to give your programs the look and feel of all the system programs and that is exactly what these references are for: Let's say you want your main window to look like the main window of QPac2. The only thing you have to do is to give the main window of your program the colour 513 (\$201). The fantastic thing is, in your system this might be a light grey, but on the system of another user it is just the colour that user has configured it to, perhaps red or dark green! You won't have to take care whether a user likes your colours: Use references for the most important colours of your program and it will look different but just fine for the user on any system. There is a simple rule to find the index of a system palette colour. Write a small program:

```
FOR i%=512 to 568
PRINT i% ! (i% MOD 256)+1 END
FOR i%
```

and you will get a list of the decimal values of the system colour palette, together with their index numbers 1 to 57. For all those colours you don't want to be changed on another system you just take the colour values of range 7) (or 1) or 2)).

From my point of view there are two conclusions:

- Programming with EasyPtr is as actual as ever before and for those who know it, it remains THE tool to write pointer programs and there is no need to learn TurboPtr.
- referencing to system colours is one of the most fascinating achievements the system colours have brought us. Together with the new colour sprites (and programs to create them), the possibility of using bitmaps (pictures,

photos etc.) the QL has been lifted to a higher level in the last months. Shouldn't we talk about a QCL (QIColourLeap) from now on :-)?

I don't hesitate to say it once more: Thank you very much, Marcel, Wolfgang (L.), Phoebus and all the others involved in this development!

BMP2SPR can be downloaded at:

<http://www.scp-paulet-lenerz.com/14mljk24/wolf/download/>

PBGCVT can be downloaded at:

<http://www.kilgus.net/soft/pngconv.zip>

QCoCo can be downloaded at:

[www.uhlich.nl/ql/qcoco.zip](http://www.uhlich.nl/ql/qcoco.zip)

If you have further questions, don't hesitate to send me an e-mail: [w.uhlig@tiscali.nl](mailto:w.uhlig@tiscali.nl)

---

## Turbo and Parameters

George Gwilt

One complaint about Turbo is that compiled programs will not run correctly either if array parameters are presented to a machine code function or procedure or if such a machine code routine alters the values of parameters.

Q\_Liberator allows this, why shouldn't Turbo allow it as well? The answer lies in the way Turbo arranges to call machine code routines. It always passes parameters by value even when the parameter is a variable. A machine code routine finds its parameters from the Basic Name Table. The entry for each parameter contains a pointer to its value. In Basic if the parameter is a simple variable the pointer points to the place where the value of the parameter is stored. If the parameter is an expression, the pointer points instead to the place where the value of the expression is stored. In the first case it is obviously possible to alter the value of the variable. Turbo, on the other hand, always treats all parameters as if they were expressions. Hence it is not possible to change the value of a variable presented as a parameter in a program compiled by Turbo.

For the same reason, a machine code extension will fail in a program compiled by Turbo if it is presented with an array parameter.

### A Solution

If a machine code extension had access to the address of a variable's value in a compiled program it could theoretically alter its value. Equally it could access the elements in an array if it knew where to look.

For its internal operations Turbo replaces Basic's Name Table by a Vector Table. This contains four bytes for every name used in a compiled program. For integer or floating point variables this long word is a pointer to the value. For arrays, the pointer is to a descriptor which in turn will contain a pointer to the set of actual values. This is similar to the array structure in Basic, but the details are different.

The function TURBO\_V, which takes as its only parameter the name of a variable, returns the absolute address in the Vector Table of the four byte entry for that variable.

Thus, in a program to be compiled by Turbo, we might put

```
1000 EXT TURBO_V(count%)
```

if EXT is a procedure expecting to alter its parameter.

The procedure itself would, of course, have to take appropriate action depending on whether it is operating in a program compiled by Turbo or not. Assuming such a test can be made, EXT would have to arrange to read in the long word parameter TURBO\_V(count%) if compiled, but the integer parameter otherwise.

In the compiled case, the value of the integer comes from the address in the Vector Table. The changed parameter is then simply fed back to the same address. This is in fact an easier process than asking the operating system to return the parameter.

The same process applies to floating point parameters.

For strings, there is a slight complication. It will not be possible for EXT to alter the size of space used by Turbo to hold the string parameter. This means that a returned string parameter is constrained to be no more than the dimensioned size of the string.

Accessing arrays in Turbo is very similar to accessing arrays in Basic. Of course, in Basic, the pointer to the descriptor will come from the parameter information in the Name Table. In a compiled program the pointer is the long word parameter pointing to Turbo's Vector Table.

Finding whether a program is compiled by Turbo can be done by examining the long word at the 34th byte from the start of the program. If this is not "Turb" it is not compiled by Turbo.

To show how all this works, an actual example is given. The extension keyword is EXT1 which can be loaded by LRESPRING file EXT1\_BIN produced by assembling EXT1\_asm.

The procedure EXT1 takes three parameters. The first is a simple integer variable, the second a simple string and the third a one dimensional integer array. EXT1 takes the first parameter as an index into the array (the third parameter), extracts the value and sets it into the first parameter. EXT1 also strips off a trailing "\_asm" found in the string parameter.

The listing of EXT1\_asm is given later. As it stands it can be assembled by either GWASL or GWASS. An example of a Basic program using it might be:

```

1 TURBO_objfil 'ram1_test_ext'
2 TURBO_taskn 'test_ext'
3 TURBO_objdat 2
4 TURBO_repfil 'ram1_err':TURBO_buffersz 2
5 TURBO_diags 'i':TURBO_struct 1:TURBO_model
6 TURBO_optim 0:TURBO_windo 0:TURBO_locstr 0
7 :
100 a%=3
110 DIM v%(6),a$(40)
120 v%(3)=-791:a$="flp1_test_asm"
130 Pr:REMark print before
140 IF COMPILED:EXT1 TURBO_V(a%), TURBO_V(a$),
    TURBO_V(v%):ELSE :EXT1 a%,a$,v%
150 Pr
160 CLOSE#3:STOP
170 :
180 DEFine PROCedure Pr
190 IF a%=3:Wind
200 PRINT#3,a%,a$
205 SUSPEND_TASK 200
210 END DEFine
220 :
230 DEFine PROCedure Wind
240 OPEN#3,scr_200x40a10x10
250 PAPER#3,7:INK#3,0:BORDER#3,2,2,4,3:CLS#3
260 END DEFine

```

Lines 1 to 6 tell Turbo how to compile the program, but are ignored by S\*BASIC. Line 140 shows the two ways of calling EXT1, depending on whether the program is compiled or not.

```

; EXT1_asm
; To show how Turbo can deal with
; (a) Altered parameters
; (b) Array parameters
;
; This sample procedure EXT1 a%,a$,v%
; sets a%=v%(a%) where v% is a one dimension integer array
; and strips off a trailing _asm (regardless of case) from a$
;

```

```

BV_VVBAS EQU $28
BV_RIP EQU $58

START LEA DEFINE,A1
MOVE.W BP_INIT,A2
JMP (A2)

DEFINE DC.W 1 one proc
DC.W EXT1-*
DC.B 4,"EXT1"
DC.W 0 end of procedures
DC.W 0 no of functions
DC.W 0 end of functions

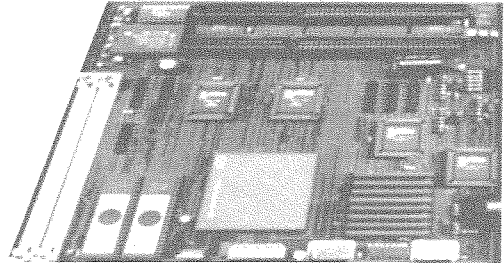
```

```

EXT1 LEA 24(A3),A0
CMPA.L A0,A5
BNE BAD_PAR —> 3 parameters needed
BSR TURBT are we in a Turbo program? . .
BNE NON_T . . no - act normally

```

# Think your own thoughts. Q60. The Super QL.



## Features

- x Q60/60 & Q60/66: 68060 CPU, 60/66 MHz, MMU+FPU
- x Q60/80: 68LC060 CPU, 80 MHz, MMU (no FPU)
- x 68060 superscalar architecture, dual execution units
- x Up to 160 BogoMIPS performance for QDOS+SMSQ/E
- x 16 to 128 MB RAM, PS/2 module sockets
- x 256 kB ROM (mainboard supports up to 1024 kB)
- x Highspeed 32 bit graphics + original QL hardware modes
- x Up to 65536 colours at 1024 x 512 pixel resolution
- x Multisync monitor output (15 pin HD connector)
- x PC Keyboard interface (DIN)
- x 20 kHz Stereo sound
- x Battery buffered clock, 2 KB nonvolatile RAM
- x Controller for 2 floppies and 2 IDE harddisks or CDROM
- x 2 Serial ports with 115200 Baud, Parallel port (on I/O card supplied with mainboard)
- x Hardware extension slot supports ISA cards
- x Fits directly into AT Minitower or other standard case
- x +5V / +12V power supply
- x No tinkering, no parts from original QL needed
- x Mainboard size 8.2 x 6.3 inch
- x Can boot in a few seconds, directly from ROM
- x Runs three different operating systems: SMSQ/E, QDOS Classic and Q60 Linux
- x New „ShoeString“ Q60 Linux distribution

## Prices

### Complete Systems

#### Q60/60 Midi Tower\*

68060 @ 60 MHz, MMU+FPU,  
64MB RAM, CD-ROM 56x,  
3.5" Floppy, 20 GB Harddisk,  
Keyboard, 3 Button Mouse,  
2 SER, 1 PAR, Stereo Sound    **£ 545.00**

#### Higher mainboard spec.

Q60/66 (68060, 66 MHz)    **+ £ 139.00**

### Extras

#### RAM

16 MB (giving 80 MB)    **£ 17.00**

64 MB (giving 128 MB)    **£ 36.00**

I/O Card (FLP,IDE,SER,PAR)    **£ 14.00**

Floppy disk drive    **£ 11.00**

#### Operating System

OS programmed on ROMs\*\*    **£ 10.00**

Q60 Linux CD    **£ 15.00**

Ethernet Card 10 Mbit/s    **£ 17.00**

#### Stereo speakers

3 boxes, including sub-woofer    **£ 19.00**

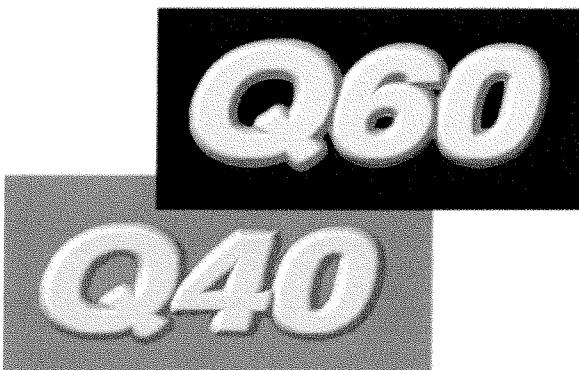
#### Preinstalled software package

QPAC1, QPAC2, FiFi, QD,  
PROWESS and much more,  
over £100 worth    **£ 59.00**

\* Fully assembled and tested! Includes support disks and manuals.

\*\* SMSQ/E and QDOS Classic available

Shipping and handling is extra. Prices may change due to semiconductor costs or exchange rates. Please note: Current SMSQ/E version supports only 16 MB out of 64 MB RAM, or 32 MB out of 80/128 MB RAM. Linux fully supports all memory configurations.



Website and technical information:

**www.q40.de**

Email: [info@q40.de](mailto:info@q40.de)

## D&D Systems

P.O. Box 5813, Ripley, Derbyshire, England DE5 9ZR

Tel. +44 (0)1773-740170, FAX +44 (0)1773-748399

After sales Tel. +44(0)1773-741164 (evenings)

Email: [sales@q40.de](mailto:sales@q40.de)

Financially assisted by a loan from QUANTA

# Take the power back in your hands.

```

;
; Here we expect pointers to Turbo's Vector Table in place of the
; parameters
;
MOVEA.W  CA_GTLIN,A2
JSR      (A2)
BNE      BAD_PAR  ——>
;
; Now pointers to the Vector Table are on the Maths Stack
;
MOVEA.L  (A1,A6.L),A2
MOVEA.L  (A2),A2           pointer to the value of a%
MOVE.W   (A2),D2           a% is now in D2.W
MOVEA.L  8(A1,A6.L),A0     Vector Table entry for v%
MOVEA.L  (A0),A0           pointer to the descriptor
TST.W    8(A0)             One dimension? . .
BNE      BAD_PAR  ——>     . . no
CMP.W    10(A0),D2         In range? . .
BGT      0_RANGE  ——>     . . no
MOVEA.L  (A0),A0           pointer to values
ADD.W    D2,D2
MOVE.W   (A0,D2.W),(A2)    set the value in a%
;
; Now the string
;
MOVEA.L  4(A1,A6.L),A0
MOVEA.L  (A0),A0           pointer to the descriptor
CMPI.W   #-1,8(A0)         no dimensions? . .
BNE      BAD_PAR  ——>     . . no - not a simple string
MOVEA.L  (A0),A1           pointer to string
SUBA.L   A6,A1             pretend to be in Basic
BSR      STR               deal with the string
END      MOVEQ    #0,D0
RTS
;
; Here is the "normal" processing
;
NON_T    MOVEA.W  CA_GTINT,A2
LEA      8(A3),A5           for 1 parameter
JSR      (A2)
BNE      BAD_PAR  ——>
MOVE.W   (A1,A6.L),D2       a% to D2.W
MOVE.W   16(A3,A6.L),D0     Name Table entry for v%
ANDI.W   #$FF0F,D0
CMPI.W   #$303,D0          an integer array? . .
BNE      BAD_PAR  ——>     . . no
MOVEA.L  20(A3,A6.L),A2     pointer to descriptor rel to VV_BAS
MOVEA.L  BV_VVBAS(A6),D0    relative start of VV_BAS
ADDA.L   D0,A2              -> descriptor rel to A6
CMPI.W   #1,4(A2,A6.L)     one dimension? . .
BNE      BAD_PAR  ——>     . . no
CMP.W    6(A2,A6.L),D2      In range? . .
BGT      0_RANGE  ——>     . . no
MOVEA.L  (A2,A6.L),A2       -> itmes rel to VV_BAS
ADDA.L   D0,A2              -> items rel to A6
ADD.W    D2,D2
ADDA.W   D2,A2              -> v%(a%) rel to A6
MOVE.W   (A2,A6.L),(A1,A6.L) set a%
MOVEA.L  A1,BV_RIP(A6)
MOVEA.W  BP_LET,A2
JSR      (A2)
BNE      BAD_PAR  ——>
;
; Now the string
;
MOVEA.L  A5,A3
LEA      8(A5),A5
MOVEA.W  CA_GTSTR,A2
JSR      (A2)
BNE      BAD_PAR  ——>

```

```

BSR      STR
BNE      END                no change in string
MOVE.L   A1,BV_RIP(A6)
MOVEA.W  BP_LET,A2
JSR      (A2)
BRA      END
;
BAD_PAR  MOVEQ    #-15,DO
RTS
O_RANGE  MOVEQ    #-4,DO
RTS

; Subroutines
;
; TURBT sets CC Z if compiled by Turbo
; Uses no registers
;
TURBT    MOVEM.L  A0-1/DO-2,-(A7)
MOVEQ   #-1,D1          This Job
MOVEQ   #0,D2          Top Job
MOVEQ   #MT_JINF,DO
TRAP    #1
CMPI.L  #"Turb",34(A0)  EQ if compiled by Turbo
MOVEM.L (A7)+,A0-1/DO-2
RTS
;
; STR does the work needed to see if the string needs shortening.
; (A1,A6.L) points to the string.
;
STR      MOVE.W   (A1,A6.L),DO      length
SUBQ.W  #4,DO          set reduced length
BMI     STEND        too short - can't be _asm
LEA     2(A1,DO.W),A2    point to last 4 characters
MOVEQ   #3,D5          count - 1
ST1     ROL.L    #8,D3          move up a byte
MOVE.B  (A2,A6.L),D3    put in the next byte
ADDQ.L  #1,A2          step to the next character
DBF     D5,ST1        count to four characters
ANDI.L  #$FFDFDFDF,D3  set 'asm' to upper case
CMPI.L  #'_ASM',D3     Was it "_ASM"? . .
BNE     STEND        . . no
MOVE.W  DO,(A1,A6.L)   set the reduced length
MOVEQ   #0,DO          mark a change occurred
STEND   RTS

```

## An Alternative Partial Solution

The obvious problem with the method described above is that it won't work unless the extensions are re-written. This may be possible in some cases but not in others. A different approach was suggested by Wolfgang Lenerz. That is to alter the code for BP\_LET so that it took account of whether the program it was in was compiled by Turbo or not. If it was, BP\_LET would find the address of the appropriate entry in the Vector Table and so determine the address of the variable. It seems logically possible for Turbo to pass the required Vector Table address to BP\_LET so a future version of Turbo might do this.

The advantage of this method is that no change would be required in the extension keywords involved, nor in the S\*BASIC programs themselves. It would mean, however, that these pro-

grams would have to be recompiled with a new version of Turbo. Also, the resulting programs would only run on appropriate future versions of SMSQ/E.

Finally, this method does not solve the problem of array parameters.

## Summary

Two methods have been described which would enable Turbo to compile programs containing calls to machine code routines which alter their parameters. The first method also allows array parameters.

The first method can be applied now to any new or revised extension keywords and to any new or revised S\*BASIC program.

The second method would not cope with array parameters and has not been implemented at the moment.

# Programming QPTR in BASIC - third part

Wolfgang Lenerz

## Reading the Pointer

Reading the pointer will enable you to get the user's response to the different possible menu actions. This, of course is a paramount part of programming in the Pointer Environment. You have the choice between two different methods of reading the pointer, with three different keywords (two of which are very similar). The first method is the most interesting, even though the second, a direct pointer read, can also be useful.

### I - Reading the pointer indirectly

This method makes our programming much easier. It is structured around the **RD\_PTR** (**ReaD PoiNTeR**) and the **RD\_PTRT** (**ReaD PoiNTeR with Timeout**) keywords. When one of these commands is used, the pointer is drawn on the screen (in the shape determined by the working definition). The user can move the pointer via the mouse or the keyboard cursor keys. The commands will only come back to the program when:

- 1) The user did (and in some cases hit) an item in a menu or an application subwindow (or used the respective selection key) and
- 2) this action did happen in this window - nothing will happen if the user clicked outside of the window.
- 3) With the RD\_PTRT keyword, a return can also be made when a timeout or "job event" occurs.

The advantage of this command seems obvious: it handles all of the changes in the pointer shape and state, notably if you have specified different pointers for application subwindows: the pointer will automatically change when it is brought over such an application subwindow. Likewise, when the pointer is moved outside of the primary window, it may change shape and become that of another window, or the default sprite (an arrow) or a sprite showing that the window underneath is not a managed window or expects keyboard input etc.

A click outside of the window is not acted upon, and, in fact the command only comes back in case of a timeout or job event (for RD\_PTRT) and when the user somehow actioned something inside of the window.

That's very practical for the programmer. When the command returns to the program, return parameters indicate what happened. Thus, there are a LOT of parameters for this command, but they are all pretty logical. We'll start with the RD\_PTR command:

### 1 - RD\_PTR

```
RD_PTR workdef, item%, subwin%, event%,  
xrel%, yrel%, liflags% {[,appflags%  
[,ctrldefx% ,ctrldefy%]]}
```

Quite a mouthful!

The parameters are the same for both RD\_PTRT and RD\_PTR, and they are as follows:

- > \* **workdef** is the window working definition. The window can be a secondary or a primary window, according to how workdef is set up. Unfortunately, when there is a primary window and a second window it is not possible to choose in which one of these you want to read the pointer: indeed, if you open a secondary window "over" a primary one (e.g. the "commands menu in the QPAC 2 FILES program) the secondary window locks the primary window over which it is pulled down and which it covers totally or partially. The primary window thus no longer is the window on top and can't read the pointer anymore.
- > \* **item%** is a returns parameter. It contains the number of the item the user hit or did, and which caused the command to return. The return mechanism is as follows: you may remember that one of the parameters for a the definition of a loose menu item or a menu application sub-window is its type (text, sprite etc) to which one adds 256 or -256: this type will then determine how the item reacts when hit/done:
  - If nothing is added to the item type , then this item acts as follows when actioned: if the item is **hit**, it just changes state - if selected it becomes available and if available it becomes selected, but it DOES NOT cause the RD\_PTR command to return. If the item is **done**, it changes state (to show that it was selected and hit) and then causes the command to return.
  - If -256 is added to the items type, both actions (hit or do) will produced the same



result, i.e. a change of state towards selected (or available if the item was already selected) and a return from the RP\_PTR command loop.

- If 256 is added to the item type, both a hit and a do will, again, have the same result: the item will cause the RD\_PTR(T) command to return to basic, but the item state will automatically be reset to available, without any programmer intervention. These last two cases (256 and -256) thus cause an "automatic return" from the item when it was hit or done.

-> \* **subwin%** is also a return parameter. It contains the number of the (application) sub-window in which the pointer was located at the time of the user action. If the pointer was not in an application subwindow but on a loose menu item or anywhere else in the window (an information sub-window, for example) then this parameter will be -1. With this, you can determine and find out whether the user clicked a loose menu item or an item in an application sub-window.

-> \* **event%**, again a return parameter, contains the "event" that caused this return. This "event" may be either the fact of actioning an item/object, or the press of any of the following keys: ESC, F1, CTRL F1, CTRL F2, CTRL F3 or CTRL F4. To each of these keypresses corresponds a certain event, and each event has a code which is thus returned in the event% return parameter. These codes are:

- 1 = DO : an item was done (ENTER)
- 2 = CANCEL: ESC was pressed
- 4 = HELP : F1 was pressed
- 8 = MOVE : CTRL F4 was pressed (move window)
- 16= SIZE : CTRL F3 was pressed (change window size)
- 32= SLEEP : CTRL F1 was pressed (make into button)
- 64= WAKE : CTRL F2 was pressed (WAKE)
- 128= HIT on an item with an automatic return.

Thus, the above keystrokes will also cause a return from the RD\_PTR(T) pointer read loop.

-> \* **xrel%** and **yrel%** are the pointer coordinates at the time when the event caused the return to the program. These coordinates are relative to the upper left corner of the window (of the application sub-window) in which the pointer was when the return occurred.

-> \* **liflags%** is the same flag array for loose items as that used for the **DR\_PULD** and **DR\_PPOS** keywords (see in an earlier instalment of this series). Remember, these flags may have a value of 0, 16 and 128. If you add one to these values when calling the **RD\_PTR** command, then the item will automatically be redrawn in the appropriate status.

-> \* **appflags%** is the same flag array for application subwindows that is used in the **DR\_PULD** and **DR\_PPOS** commands which we treated in an earlier instalment of this series. Just like for loose menu items, if you set any value of these flags arrays to the status +1, the items will be redrawn automatically upon entering this command.

-> \* **ctrldefx%** and **ctrldefy%** are the application sub-window control definition arrays.

Using this command is pretty easy because it only causes a return for well defined events. It can get included in a pointer read loop which will be about as follows:

```
REPEAT loop
RD_PTR <parameters>
post=item%:REMark SElect on floats only in QDOS
SElect ON subwin
  --1      : rem loose menu item
    SElect ON post
      =1:do_this
      =2:do_that
      =3:something_else
      ... etc...
    END SElect
  =0      : rem click in first menu appsub wdw
    SElect ON post
      ....
    END SElect
  END SElect
END REPEAT loop
```

Thus, one reads the pointer and when the return was made, one uses the subwindow and the item to determine, first, in what subwindow the event occurred and, second, what action should be taken for this event.

## 2 - RD\_PTRT, timeouts and job events

The **RD\_PTRT** keyword is pretty similar to the **RD\_PTR** keyword. Both use mostly the same parameters, except that the **RD\_PTRT** keyword has one additional parameter, a timeout, as follows:

```
RD_PTRT workdef, item%, subwin%, event%,
timeout%, xrel%, yrel%, liflags%
{[,appflags% [,ctrldefx% ,ctrldefy%]]}
```

The **workdef**, **item%**, **xrel%**, **yrel%**, **liflags%**, **appflags%**, **ctrldefx%** and **ctrldefy%** parameters are the same as for **RD\_PTR** and thus don't need to be described here anymore.

There are two changes with respect to the **RD\_PTR** keyword:

First of all, there is an additional **timeout%** parameter. With this you can indicate that you also want a return from the keyword after a certain time. In usual QL fashion, the timeout is given in 1/50th of a second (me thinks, in North America it is in 1/60th of a second).

Thus, you can also make sure that you return from the read pointer loop after a certain period of inactivity. Mind, though, that the return will be made either because of a "normal" event (including "job events, which I'll explain below) or because of a timeout - whatever comes first!

When a return from a timeout occurs, the event parameter is set to -1, which is a value it doesn't normally have. This allows you to distinguish between normal events and a timeout.

Speaking of the **event%** parameter, this has been modified a bit. It still has all of the functions as for the **RD\_PTR** keyword, but has been extended. You can also use it as an entry parameter for the **RD\_PTRT** command, to pass it some "job events" on which the keyword will also return.

Job events are a relatively recent addition to the Pointer Environment. They are an easy, legal and (now) documented way for one job (program) to communicate with another. One program can send another an "event". The other program receives the event through the read pointer loop. There are 8 events, contained in one byte, each bit representing one event.

Sending an event is pretty easy, and uses the

typical TK II fashion of determining a job:

```
SEND_EVENT job_id,event
(the job_id is a composite number: job_tag *
65536 + job_nbr),
or
SEND_EVENT "job_name", event
or
SEND_EVENT job_nbr,job_tag,event
```

For example:

```
SEND_EVENT "Quill",3
will send events 1 and 2 to Quill. This wouldn't
mean a lot, since Quill isn't equipped to receive
events, but it could be done.
```

When the program is a pointer program, it will receive the event through the pointer read loop. In S\*Basic, this event may cause a return from the pointer read loop: "May" not "will" - at least not necessarily.

Indeed, the **event%** parameter will indicate, on entry to the command, what job events the program is ready to receive. If the **event%** parameter is 0, it is not ready to receive any event. If **event% = 1 \* 256**, it is ready to receive event 1, if it is **3 \* 256**, the program is ready to receive events 1 & 2 and so on.

As you can see, the event is passed in the high byte of the **event%** word, thus just multiply the events to indicate by 256. There is one problem: if you want to indicate that you are ready to receive all 8 events, you would normally have to pass **255\*256**. This will cause an error, so use -1 instead.

On return from the read pointer loop, the job events are contained in the upper byte of the **event%** word.

There is also a way to get an event without reading the pointer:

```
result%=WAIT_EVENT (events% [,timeout%])
```

This will wait for **timour%** ticks (if this parameter is not passed, it waits forever) until one of the **evnts%** passed on entry happens. The event(s) are returned in **result%**

This was the easy way to read the pointer. Next time, we'll look at a more circumvallated way of doing this.

QL Forever!

# JOCHEN MERZ SOFTWARE

**Im stillen Winkel 12 D-47169 Duisburg**  
**Tel. 0203 502011 Fax 0203 502012**  
**http://smsq.j-m-s.com smsq@j m s.com**

## Sooo much colour...

... talk in this issue - can you imagine not having it?

As mentioned in earlier issues, please send in your master disks for proof of purchase. They will be returned unmodified to you, so that you still have access to the older versions. For every product listed below, you will get a new disk with the current high-colour updated software!

I suggest you order the Colour Utility disk one time if you upgrade to high-colour, as it contains utilities and samples how to change colours, themes and even sprites within a theme (thanks to Roy Wood and Marcel Kilgus).

### Update & Upgrades (für Hi-Colour)

<b>QPC2 Version 3 - Upgrade from QPC2 Version 2</b>	<b>EUR 20,40</b>
<b>QPC2 Version 3 - Upgrade from QPC2 Version 1</b>	<b>EUR 44,90</b>
<b>QPC2 Version 3 - Upgrade from QPC1</b>	<b>EUR 69,90</b>
<b>SMSQ/E Version 3 - Upgrade for (Super)GoldCard with Aurora-Colour Drivers</b>	<b>EUR 34,90</b>
<b>QPC2 Version 3 - Update from QPC2 Version 3 to current Version</b>	<b>EUR 0,80</b>
<b>SMSQ/E Version 3 - Update from ATARI, (Super)GoldCard or QXL</b>	<b>EUR 0,80</b>
<b>text87patch</b> (see description under "News")	<b>EUR 12,00</b>

We recommend the Colour Utility Disk if you upgrade to SMSQ/E Version 3! **EUR 0,80**

<b>QD2003 - Upgrade from QD98</b>	<b>[VB.01]</b>	<b>EUR 14,90</b>
<b>QD2003 - Upgrade from previous versions</b>	<b>[VB.01]</b>	<b>EUR 29,90</b>
<b>QSpread2003 - Upgrade from QSpread2001</b>	<b>[V4.00]</b>	<b>EUR 14,90</b>
<b>QSpread2003 - Upgrade from V1</b>	<b>[V4.00]</b>	<b>EUR 39,90</b>
<b>FiFi II - Update from V4.xx</b>	<b>[V4.30]</b>	<b>EUR 0,80</b>
<b>FiFi II - Upgrade from FiFi V1, 2 or 3</b>	<b>[V4.30]</b>	<b>EUR 9,90</b>
<b>WINED - Update</b>	<b>[V1.26]</b>	<b>EUR 0,80</b>
<b>EPROM Manager - Update</b>	<b>[V3.02]</b>	<b>EUR 0,80</b>
<b>QPAC I - (Important! New Sysmon for SMSQ/E V3)</b>	<b>[V1.09]</b>	<b>EUR 0,80</b>
<b>QPAC II - Update with some useful improvements</b>	<b>[V1.42]</b>	<b>EUR 0,80</b>
<b>QSUP - Update</b>	<b>[V4.00]</b>	<b>EUR 0,80</b>
<b>QMAKE - Update</b>	<b>[V4.31]</b>	<b>EUR 0,80</b>

**Prices include NEW DISK PER PRODUCT AND POSTAGE PER DISK for deliveries to Europe. For delivery to USA, add EUR 0,20 per disk.**

**Please add one-time packaging charge 2 EUR for Europe, 4 EUR for USA.**

# QLTdis part 11

Norman Dunbar

Well, my enforced period of redundancy is over (for now) as I am once more working as an Oracle DBA. This time I'm on a 3 month contract (finishing in March 2004) at a certain large bank in Knutsford, Cheshire. This means that I'm typing this in my 'digs' in a small village B&B on my laptop. I think I started this whole series on a laptop in Manchester at Oracle training, and here I am, doing a similar thing five or six years on. Can it really be that long? (It seems longer!)

Well, it looks like the register list decoding routine required by the MOVEM instructions have got me beaten. The code I converted from SuperBasic doesn't actually work (properly).

To this end, and after much wailing and gnashing of teeth, I've had to admit defeat on decoding the register lists using code to check the various bits. It is unfortunate, but as soon as I tracked down one problem, other bit's (no pun intended) I'd already fixed stopped working again. The comes a time when enough is enough.

I've redesigned the register list decoder to use a pair of tables. The first holds an offset into the second table as a word of data. The second table holds all the instructions decoded according to the bit value in a byte. The full code is shown below, but here is a small extract of what I'm on about.

```
reg_off equ *
r_1 dc.w mml_1-reg_table
r_2 dc.w mml_2-reg_table
r_3 dc.w mml_3-reg_table
```

That is the first offset table. It simply holds one word in each element. The word is the offset into the reg\_table where the decoded register list string can be found for the specific value of the byte holding the register list 'bitmap'. The reg\_table looks like this:

```
reg_table equ *
mml_1 dc.w mml_2-mml_1-2
dc.b 'D0'
mml_2 dc.w mml_3-mml_2-2
dc.b 'D1'
mml_3 dc.w mml_4-mml_3-2
dc.b 'D0-D1'
```

And so on and on for 256 entries in each table. Now, I'm not fond of typing all this stuff in line after line, so here's a small program to build up the first table followed by the second table as much as possible.

```
1000 REMark Alter the file name in the
      following line to suit where you keep
1005 REMark your source files.
1010 :
1015 OPEN #3,'win1_source_dis_reglist_asm'
1020 :
1025 REMark Build reg_off table first
1030 PRINT #3,'reg_off equ *'
1035 FOR Label = 1 TO 255
1040 PRINT #3, 'r_'; Label; ' dc.w mml_';
      Label; '-reg_table'
1045 END FOR Label
1050 PRINT #3
1055 :
1060 REMark Then build reg_table table (!)
1065 PRINT #3,'reg_table equ *'
1070 FOR Label = 1 TO 255
1075 PRINT #3, 'mml_'; Label; ' dc.w
      mml_'; Label+1; '-mml_'; Label; '-2'
1080 PRINT #3, ' dc.b'; " 'D'"
1085 END FOR Label
1090 PRINT #3, 'mml_256 equ *'
1095 :
1100 REMark All done
1105 PRINT #3
1110 CLOSE #3
```

That should reduce the typing somewhat. All you have to do now is load the file into your favourite text editor and fill in the register lists as per the full listing below. I'm a big fan of getting the QL to do the hard work for me as much as I can.

Actually, if you wanted to, and didn't mind the format, you could write a function which takes a byte value parameter and returns a string holding the decoded register list. Sounds complicated but it isn't.

The returned string could look like 'D1,D2,D3,D7' instead of 'D1-D3,D7'. If you did that, you wouldn't need to type anything in for the reglist\_asm file. Actually, it could be worse, the decoded instruction could be shown as:

```
movem.l (a7)+,d0,d1,d2,d3,d4,d5,d6,d7/
a0,a1,a2,a3,a4,a5,a6
```

rather than the much more user friendly:

```
movem.l (a7)+,d0-d7/a0-a6
```

However, think of all that typing you won't have to do :o)

Try this function for size:

```
2000 DEFine FuNction RegList$(bitmap)
2005   LOCal RList$, bit
2010   RList$ = '' : REMark no space between
        quotes.
2015   FOR bit = 0 to 7
2020   IF bitmap && 2^bit THEN
2025     RList$ = RList$ & ',D' & bit
2030   END IF
2035   END FOR bit
2040   RETurn RList(2 TO) : REMark strip off
        the leading comma
2045 END DEFine RegList$
```

With this in place, change line 1080 to the following:

```
1080 PRINT #3,, 'dc.b', ""&RegList$(label)&""
```

In case the quotes are funny in the printed magazine, they are double single double, at the start and end of the call to RegList\$(label). Ok enough of this, onwards with the code description.

We need to use two tables because of the problem that the decoded register lists are variable length strings. In reverse order, the second table holds a list of 255 register list strings in normal QDOSMSQ format with a word count followed by the bytes of the string.

The first table, reg\_off, is simply a table of 255 words. Each word is the offset from the start of

the second table to the start of the decoded string for the appropriate bitmap.

We will of course require a table of data register lists and one of address register lists, won't we? So why does the code above only build a single table of data register strings?

You may well be thinking ahead and saying to yourself, 'does this mean I have to type in another table for the Address registers?'. Thankfully, the answer is 'no'. The same table will be used, but when we are decoding a data register list, the decoded string is appended to the output buffer as is.

If, on the other hand we are decoding a list of address registers, then we will replace all the 'D' characters with an 'A' at some point before or after it gets appended to the buffer.

What we will do is copy the register list string into the input buffer (which we are not using at this point in time) then replace the every 'D' with an 'A'.

Once we have done this, the resulting string is copied from the input buffer to the output buffer.

The following code replaces everything in the original type 27 routines.

---

```
*-----*
* TYPE 27 - the MOVEM instructions
*-----*
dtype_27    btst      #6,d7          ; Check the size specifier
            beq.s     t27_word       ; Clear = .W
            bsr       ell            ; Add '.L' size details
            moveq     #4,d5          ; Set the op-code size too
            bra.s     t27_all        ; Skip word stuff

t27_word    bsr       uu             ; Add '.W' size details
            moveq     #2,d5          ; Set the op-code size

t27_all     bsr       space          ; Add a space to the buffer
            move.w    (a6)+, -(a7)   ; Stack the register list word
            btst     #10,d7          ; Set = Mem->Reg
            beq.s     t27_r2m        ; Clear = Reg->Mem

t27_m2r     bsr       eff_addr       ; Effective address is extracted
            bsr       comma          ; Then a comma
            move.w    (a7)+, d2      ; Get the register list word
            ror.w     #8,d2          ; Rotate the high byte -> low byte
            move.b    d2,d4          ; Copy the (old) high byte

t27_addr    bsr       addr_reg       ; Extract the address registers
            bne.s     t27_mask        ; No address registers to copy, so don't !
            lea      buffer, a1      ; Decoded address register list is here
            bsr.s     t27_append     ; Append to decoded instruction

t27_mask    ror.w     #8,d2          ; (old) low byte back again
```

\* D2.B holds the data register mask now. D4.B still holds the address register  
 \* mask. T27\_SLASH will determine whether a slash is required in the fully  
 \* decoded register list.

```

        bsr          t27_slash          ; Add a slash if required

t27_data  move.b     d2,d4              ; Get the data register list
        bsr          data_reg          ; Extract them
        bne.s       t27_skip          ; Nothing to copy, so don't !
        movea.l     a3,a1             ; Data register list string
        bsr.s       t27_append        ; Add to output buffer
t27_skip  bra        p_hex             ; Finished with mem -> reg

```

\* At this point we are processing a register to memory MOVEM instruction. We  
 \* have to do some extra processing to determine if pre-decrement mode is being  
 \* used and if so, reverse the order of the bits in the register bitmap. Why  
 \* Motorola chose to have one single instruction with the bits back to front, I  
 \* don't know !

```

t27_r2m   move.w     (a7)+,d2          ; Fetch the register list word
        andi.b      #$38,d0          ; Extract the mode bits in the op-code
        cmpi.b      #$20,d0          ; Mode = '100' = pre-decrement
        bne.s       t27_notpd        ; Not pre-decrement
        bsr          swap_d2          ; Reverse the bits in D2 for pre-dec

t27_notpd  move.b     d2,d4            ; Data registers in low byte
        bsr          data_reg          ; Extract them
        bne.s       t27_nodata       ; No data registers to copy, so don't !
        movea.l     a3,a1             ; String location
        bsr.s       t27_append        ; Append data reg list to output buffer

t27_nodata  ror.w     #8,d2           ; Shift address register mask to low byte

```

\* D2.B holds the address register mask now. D4.B still holds the data register  
 \* mask. T27\_SLASH will determine whether a slash is required in the fully  
 \* decoded register list.

```

        bsr          t27_slash          ; Add a slash if required
        move.b      d2,d4              ; Address register list bitmap
        bsr          addr_reg          ; Extract address register list
        bne.s       t27_comma         ; No address registers to copy.
        movea.l     a3,a1             ; String location (from address)
        bsr.s       t27_append        ; And append the address register list

t27_comma  bsr          comma          ; Comma required
        bsr          eff_addr          ; And finally, the effective address
        bra         p_hex             ; Finished with reg -> mem

```

---

\*  
 \* T27\_SLASH - adds a slash if required  
 \*  
 \* D4.B = address (or data) register list mask  
 \* D2.B = data (or address) register list mask  
 \*  
 \* A slash will be required if both are non-zero.

---

```

t27_slash  tst.b      d2                ; No slash if zero
        beq.s       t27_nosl2        ;
        tst.b      d4                ; No slash if zero
        beq.s       t27_nosl2        ;
        bsr          slash            ; Add a '/'
t27_nosl2  rts

```

---

\*  
 \* T27\_APPEND - append the decoded register list to the decoded instruction  
 \* buffer.  
 \*  
 \* D6.W holds the current size of the output buffer and A5.L is the pointer  
 \* to the first free character in said buffer. This means that the str\_append  
 \* sub-routine is not useful here until we do some jiggery-pokery with the buffer.

# RWAP Services NEWS!

## New Websites!

We are proud to present our new websites! Starting September, we have separated our business in three separate websites.

You can find them at:

<<http://www.rwapadventures.com/>> (Adventure Games)  
<<http://www.rwapervices.co.uk/>> (General Site) and  
<<http://www.rwapsoftware.co.uk/>> (Software)

Our old address: <<http://hometown.aol.co.uk/RWAPSoftware/>> is of course still functioning but will be deprecated in the future.

## New Products!

**QWORD 2.0** NOW WITH DIGITAL  
SOUND ON QPC2!

The wait is now over! Q-Word version 1 is finally available!

Platforms:

QPC/QXL, Q40/Q60, Aurora (with SGC)

Prices:

All versions without P-Word £20.00p  
All versions with P-Word £30.00p

Notes:

Q-Word DOES NOT require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.

The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2

## Quantum Leap ED Drives

After many years of unavailability, here they are again! These are high quality Mitsubishi constructed/IBM badged drives with full warranty.

Unlike previously sold ED drives, these do not require FLP\_JIGGLE and have no problems formatting 720K disks. However for the latter a switch is included with the cable.

ED Bare unit (no cable) £ 29.99p  
ED Boxed unit (complete with cable/ PSU) £ 98.99p  
Single unit Cable (with switch) £ 4.99p  
Dual unit Cable (with two switches) £ 5.99p  
(More options available, contact us for details. Also available mass quantities of ED and DD disks!)

## Talent Games for Windows

For QLers that run Windows or with incompatible hardware for Talent Games, we now have re-released these adventures so that they can run on your Windows-equipped PC. No Emulator, floppies, microdrive backups etc. required, just a one-click install! Of course the full QL line is still available! (See side column)

Talent Games for Windows ea. £ 10.00p  
(Each Game includes a runtime installation of QLAY-2 by Jimmy Montesinos)

## RWAP Services

35 Chantry Croft, Kinsley, Pontefract, West Yorkshire, WF9 5JH, United Kingdom Tel: +44 1977 610509 (From the UK dial: 01977 610509)  
Website: <http://www.rwapservices.co.uk>

We accept:



(For PayPal please add 4% to the total price. Cheques in £ sterling made payable to R. Mellor)

## Old Favourites!

### Utilities

Sidewriter v1.08	£ 10.00p
Landscape Printing (EPSON printers)	
ImageD v1.03	£ 10.00p
3D object generator	
Q-Help v1.06	£ 10.00p
Superbasic On-Screen help system	
Q-Index v1.05	£ 5.00p
Keyword-to-topic finder	
ProForma ESC/P2 Drivers v1.04 for ProWeSs	£ 8.00p
Printer Driver	

### Applications

Flashback SE v2.03 (upgrade only)	£ 5.00p
Database	
QL Cash Trader v3.7	£ 5.00p
Accounting/Finance	
QL Payroll v3.5	£ 5.00p
Accounting/Finance	
QL Genealogist v3.26	£ 20.00p
Genealogy	
Genealogy for Windows	£ 50.00p
QL Genealogist to Windows version upgrade	£ 25.00p
QL Cosmos v2.04	£ 5.00p
Planetarium	
Q-Route v2.00	£ 25.00p
Route Finding	
Upgrade from v1.xx	£ 5.00p
Britain map v1.11	£ 2.00p
BIG Britain map (needs 2Mb) v2.03	£ 5.00p
Various Britain Area maps (ask for details)	ea. £ 2.00p
Ireland map v1.00	£ 5.00p
Belgium map v1.01	£ 2.00p
Catalonia map v1.03	£ 2.00p
P-Word UK English Dictionary (500.000 words!)	£ 15.00p
Dictionary	

### Leisure

Return to Eden v3.08	£ 10.00p
Adventure	
Nemesis MkII v2.03	£ 8.00p
Adventure	
The Prawn v2.01	£ 8.00p
Adventure	
Horroraday v3.1	£ 8.00p
Adventure	
West v2.00	£ 5.00p
Adventure	
The Lost Kingdom of Zkul v2.01	£ 5.00p
Adventure	
All 6 games above	£ 25.00p
D-Day MkII v3.04	£ 10.00p
Strategy/War Simulation	
Grey Wolf v1.08	£ 8.00p
Graphical Submarine Simulation	
War in the East MkII v1.24 (upgrade only)	£ 5.00p
Strategy/War Simulation	
Open Golf v5.20	£ 8.00p
Sports Simulation	
QuizMaster II v2.07	£ 5.00p
Quiz	
Stone Raider II v2.00	£ 5.00p
Arcade Game	
Hoverzone v1.2	£ 5.00p
Arcade Game	
Deathstrike v1.5	£ 5.00p
Arcade Game	
Flightdeck v1.0	£ 10.00p
Flight Simulation	
All 6 games above (Open Golf, QuizMaster II, Stone Raider II, Hoverzone, Deathstrike and Flightdeck)	£ 28.00p

### Notes on Software requirements

The following programs have a minimum SGC card requirement: P-Word, Qword, Big Britain MAP for Q-Route

```

* On entry, A1 contains the address of the source string as this can be
* either an entry in the reg_table (for data register lists) or the input
* buffer if we are decoding the address register lists.
*
* Corrupts D3, A2 and A1 - all of which are working registers and we
* don't care !
*

```

```

t27_append  move.w    d4,-(a7)          ; Preserve the register list mask
           lea      output,a2        ; Destination string
           move.w   d6,(a2)          ; Save the string size correctly
           add.w    (a1),d6          ; Update resulting string length
           bsr     str_append        ; Append the string to the buffer
           move.w   (a7)+,d4         ; Restore the register list bitmap
           rts

```

The following line should be added to your source code. It pulls in the file we created above containing all the decoded register lists. By generating it into a separate file as automatically as possible, then including it, we have less typing to do to get it all included in the source code.

I've added the following just after the end of the type 31 code and just above the cond\_code routine which decodes the various condition codes.

```

*-----
* Insert the register list decoding tables next
*-----
        in win1_source_QLTdis_reglist_asm

```

Some changes are required in the addr\_reg and data\_reg routines as well. Change both of those routines to the following code.

Note, if you decided in the SuperBasic code above that your registers would be in lower case, then in the following code, at label ar\_d2a, you'll need to replace the 'D' in cmpi #'D',(a3) with a 'd' and change the 'A' to 'a' two lines further on.

```

*-----
* ADDRESS Register list routine. Returns the address of the decoded address
* register list in A3.L. On entry, requires a register list bitmap byte in D4.
* Also returns D0 = 0 and Z set if no errors, or Z unset and D0 = ERR_BP if
* there is nothing in the register list to decode.
*-----
addr_reg  tst.b     d4                ; Nothing to do if D4 is zero
         beq.s    ar_nothing        ; Thought so !
         move.w   d4,-(a7)          ; Preserve the register list bitmap
         bsr.s    reg_list          ; Returns A3 as a reg list string pointer

```

```

* At this point, A3 holds the address of the decoded string but it has DATA
* register names in it. We need to copy it to the input buffer, convert from
* DATA to ADDRESS registers by replacing all 'D' with 'A' then copy the
* resulting string to the output buffer.

```

```

         movem.l  a1-a2,-(a7)        ; Save the workers
         movea.l  a3,a1              ; From address
         lea     buffer,a2           ; To address
         bsr     str_cpy             ; Copy the string to the input buffer
         movem.l (a7)+,a1-a2        ; Restore the workers

         lea     buffer,a3           ; Start of string = word count
         move.w  (a3)+,d4            ; Size of string
         bra.s   ar_d2a_end         ; Skip

ar_d2a   cmpi.b  #'D',(a3)           ; Found a 'D' ?
         bne.s   ar_d2a_next        ; Fraid not
         move.b  #'A',(a3)          ; Yes, change it

```



```

ar_d2a_next adda.l    #1,a3          ; Next character in buffer
ar_d2a_end  dbra     d4,ar_d2a      ; Do rest of buffer

                move.w    (a7),d4    ; Restore register list bitmap
                clr.l     d0         ; No errors
                bra.s     ar_exit    ; Finished.

ar_nothing  moveq     #ERR_BP,d0    ; Nothing to decode = bad parameter
ar_exit     tst.b     d0            ; Set or clear Z accordingly
                rts                ; Return with A3 -> register list string

```

```

*-----
* DATA Register list routine. Returns the address of the decoded data register
* list in A3.L. On entry, requires a register list bitmap byte in D4.
* Also returns D0 = 0 and Z set if no errors, or Z unset and D0 = ERR_BP if
* there is nothing in the register list to decode.
*-----

```

```

data_reg    tst.b     d4            ; Nothing to do if D4 is zero
                beq.s     dr_nothing ; Thought so !
                bsr.s     reg_list  ; Returns A3 as a reg list string pointer
                clr.l     d0         ; No errors
                bra.s     dr_exit    ; Finished.

dr_nothing  moveq     #ERR_BP,d0    ; Nothing to decode - bad parameter
dr_exit     tst.b     d0            ; Set or clear Zero flag accordingly
                rts

```

The old code at routine reg\_list should be deleted and the following inserted in its place. All the old code from reg\_list down to (and including) label rl\_done should be replaced with the following:

```

*-----
* Register list routine. This returns A3 as a pointer to a register list string
* according to the value of the register list bitmap passed in D4.B.
*
* Because there is no entry in the table for zero, we must adjust the entry we
* want to account for this. The register list for a bitmap with the value of
* one is in position zero of the reg_off table and so on. We must subtract 1
* from the bitmap value to get our proper table entry number in the reg_off
* table. D4.B is guaranteed to be between 1 and 255.
*-----
reg_list     lea        reg_off,a3   ; Offset table in A3
                subq.b   #1,d4      ; Adjust to correct table entry
                ext.w    d4          ; D4 needs to be word sized
                lsl.w    #1,d4      ; Double D4 - each entry is two bytes
                adda.w   d4,a3       ; A3 points to the correct offset word
                move.w   (a3),d4     ; Extract the offset
                lea.l    reg_table,a3 ; Start of the decoded reg list table
                adda.w   d4,a3       ; Correct string address in A3
                rts

```

The diss\_asm file should now be saved. The remainder of the code is the reglist\_asm file created by the SuperBasic procedure at the beginning of this article.

If you decided to go for the 'no typing at all' version, skip to the bottom of the code - you don't need to type anything. If you decided to be a martyr to the cause, then read on ...

The file reglist\_asm, generated in part above, should look like the code below. To save space, I will not be importing the entire reg\_off (register offset) table here. There is no need as the table was generated by the small program above. Suf-

fice to say that it starts at label r\_1 and stops at r\_255 with every entry in between catered for. To save wasting paper, I'm only showing the first and last 5 entries here.

As mentioned above, each entry is a single word of data representing the offset of the decoded string - the register list - for each register list bitmap byte in a MOVEM instruction. There is no list if the value is zero, so we don't have an entry for zero - in case you were wondering.

```

reg_off equ *
r_1      dc.w    mml_1-reg_table
r_2      dc.w    mml_2-reg_table
r_3      dc.w    mml_3-reg_table

```

```

r_4      dc.w      mml_4-reg_table
r_5      dc.w      mml_5-reg_table

.....

r_250    dc.w      mml_250-reg_table
r_251    dc.w      mml_251-reg_table
r_252    dc.w      mml_252-reg_table
r_253    dc.w      mml_253-reg_table
r_254    dc.w      mml_254-reg_table
r_255    dc.w      mml_255-reg_table

```

Stop typing for a bit!

That's the easy bit done. Now for some serious typing (if this isn't on a floppy disc somewhere in the magazine that is!)

We have the next table, this is the actual register list table, into which the table above provides us with an entry offset. This table has been partially written by the small SuperBasic routine above, but you still have to fill in the decoded strings yourself. Thankfully, you don't have to do as much typing as at first imagined. Currently, it looks similar to this:

```

reg_table      equ *
mml_1  dc.w    mml_2-mml_1-2
        dc.b    'D'
mml_2  dc.w    mml_3-mml_2-2
        dc.b    'D'
mml_3  dc.w    mml_4-mml_3-2
        dc.b    'D'

....

mml_254 dc.w    mml_255-mml_254-2
        dc.b    'D'
mml_255 dc.w    mml_256-mml_255-2
        dc.b    'D'
mml_256 equ *

```

As ever, I've got the assembler to count up the string sizes for me, that's what computers are for after all. The calculation fills in the word length of the decoded register list string so that I don't have to remember to count up the size of each string and physically type it in. I'm a bit lazy by nature!

Unfortunately, we still have 255 different register list strings to type in, but at least the first letter is done for us. Note that the code in the address register list decoding routine will scan the string and replace all 'D' with 'A' so if you prefer lowercase register names, change it here and in the code above.

Lets get it over with. I've split the following table up into groups of 10 so that it is a bit easier to

follow in the listing. Hopefully, Jochen will not reformat it as one big chunk. I don't know about you, but I used to hate typing in all that hex data (before I had an assembler) where the author (or publisher) had set it out in one huge block of data.

Ok, you can start typing again, remember, all you need is the bit in the second line of each entry, the rest is already there for you.

```

reg_table      equ *
mml_1  dc.w    mml_2-mml_1-2
        dc.b    'D0'
mml_2  dc.w    mml_3-mml_2-2
        dc.b    'D1'
mml_3  dc.w    mml_4-mml_3-2
        dc.b    'D0-D1'
mml_4  dc.w    mml_5-mml_4-2
        dc.b    'D2'
mml_5  dc.w    mml_6-mml_5-2
        dc.b    'D0,D2'
mml_6  dc.w    mml_7-mml_6-2
        dc.b    'D1-D2'
mml_7  dc.w    mml_8-mml_7-2
        dc.b    'D0-D2'
mml_8  dc.w    mml_9-mml_8-2
        dc.b    'D3'
mml_9  dc.w    mml_10-mml_9-2
        dc.b    'D0,D3'
mml_10 dc.w    mml_11-mml_10-2
        dc.b    'D1,D3'

mml_11 dc.w    mml_12-mml_11-2
        dc.b    'D0-D1,D3'
mml_12 dc.w    mml_13-mml_12-2
        dc.b    'D2-D3'
mml_13 dc.w    mml_14-mml_13-2
        dc.b    'D0,D2-D3'
mml_14 dc.w    mml_15-mml_14-2
        dc.b    'D1-D2-D3'
mml_15 dc.w    mml_16-mml_15-2
        dc.b    'D0-D3'
mml_16 dc.w    mml_17-mml_16-2
        dc.b    'D4'
mml_17 dc.w    mml_18-mml_17-2
        dc.b    'D0,D4'
mml_18 dc.w    mml_19-mml_18-2
        dc.b    'D1,D4'
mml_19 dc.w    mml_20-mml_19-2
        dc.b    'D0-D1,D4'
mml_20 dc.w    mml_21-mml_20-2
        dc.b    'D2,D4'

mml_21 dc.w    mml_22-mml_21-2
        dc.b    'D0,D2,D4'
mml_22 dc.w    mml_23-mml_22-2
        dc.b    'D1-D2,D4'
mml_23 dc.w    mml_24-mml_23-2
        dc.b    'D0-D4'
mml_24 dc.w    mml_25-mml_24-2
        dc.b    'D3-D4'
mml_25 dc.w    mml_26-mml_25-2
        dc.b    'D0,D3-D4'
mml_26 dc.w    mml_27-mml_26-2
        dc.b    'D1,D3-D4'
mml_27 dc.w    mml_28-mml_27-2
        dc.b    'D0-D1,D3-D4'
mml_28 dc.w    mml_29-mml_28-2

```

## The Bulletin Boards System - Help required!

Well, the BBS or Mailbox is giving me a headache, for quite a while now. I just don't find the time to maintain it, and although I see the machine every day and check the faxes (they are received by the Mailbox system too) I hardly ever log in. Whenever I start to log in regularly, there is no mail, and I give up after some time.

Sorry, but that's the way it is. E-Mail has completely taken over the BBS message system.

But that's not the only problem - BBS maintenance in general is something I rarely find the time. News spread much easier over the net too.

The only real reason so far was to have to BBS running for free software updates and upgrades. But I am also thinking about finding a better solution for this too, where the net comes into play.

The problem is: I do not know how to move the functionality of the group download restrictions onto the web.

What I need is some kind of ASP script or so, where users can register and log in via username and password (provided by me) and where I can assign every user the rights to access the download of individual files (e.g. specify user X can download QD and QSpread updates, user Y can download FiFi and QD etc.)

The complete requirements (traffic limits, download limits, how to ensure that the downloadable files are not publicly available) need to be discussed, of course.

I am not a web expert, so I am wondering if any of you, the readers, have more knowledge and can help?

More and more users ask for web updates, and I think it is about time to move from the BBS to an internet based update service, if that is possible somehow.

Well, I am somehow prepared to do it now, but I don't know how ... maybe somebody else has got an idea?

Any feedback please to [SMSQ@j-m-s.com](mailto:SMSQ@j-m-s.com)

mml_29	dc.b	'D2-D4'	mml_62	dc.w	mml_63-mml_62-2
	dc.w	mml_30-mml_29-2		dc.b	'D1-D5'
	dc.b	'D0,D2-D4'	mml_63	dc.w	mml_64-mml_63-2
mml_30	dc.w	mml_31-mml_30-2		dc.b	'D0-D5'
	dc.b	'D1-D4'	mml_64	dc.w	mml_65-mml_64-2
				dc.b	'D6'
mml_31	dc.w	mml_32-mml_31-2	mml_65	dc.w	mml_66-mml_65-2
	dc.b	'D0-D4'		dc.b	'D0,D6'
mml_32	dc.w	mml_33-mml_32-2	mml_66	dc.w	mml_67-mml_66-2
	dc.b	'D5'		dc.b	'D1,D6'
mml_33	dc.w	mml_34-mml_33-2	mml_67	dc.w	mml_68-mml_67-2
	dc.b	'D0,D5'		dc.b	'D0-D1,D6'
mml_34	dc.w	mml_35-mml_34-2	mml_68	dc.w	mml_69-mml_68-2
	dc.b	'D1,D5'		dc.b	'D2,D6'
mml_35	dc.w	mml_36-mml_35-2	mml_69	dc.w	mml_70-mml_69-2
	dc.b	'D0-D1,D5'		dc.b	'D0,D2,D6'
mml_36	dc.w	mml_37-mml_36-2	mml_70	dc.w	mml_71-mml_70-2
	dc.b	'D2,D5'		dc.b	'D1-D2,D6'
mml_37	dc.w	mml_38-mml_37-2			
	dc.b	'D0,D2,D5'	mml_71	dc.w	mml_72-mml_71-2
mml_38	dc.w	mml_39-mml_38-2		dc.b	'D0-D2,D6'
	dc.b	'D1-D2,D5'	mml_72	dc.w	mml_73-mml_72-2
mml_39	dc.w	mml_40-mml_39-2		dc.b	'D3,D6'
	dc.b	'D0-D2,D5'	mml_73	dc.w	mml_74-mml_73-2
mml_40	dc.w	mml_41-mml_40-2		dc.b	'D0,D3,D6'
	dc.b	'D3,D5'	mml_74	dc.w	mml_75-mml_74-2
				dc.b	'D1,D3,D6'
mml_41	dc.w	mml_42-mml_41-2	mml_75	dc.w	mml_76-mml_75-2
	dc.b	'D0,D3,D5'		dc.b	'D0-D1,D3,D6'
mml_42	dc.w	mml_43-mml_42-2	mml_76	dc.w	mml_77-mml_76-2
	dc.b	'D1,D3,D5'		dc.b	'D2-D3,D6'
mml_43	dc.w	mml_44-mml_43-2	mml_77	dc.w	mml_78-mml_77-2
	dc.b	'D0-D1,D3,D5'		dc.b	'D0,D2-D3,D6'
mml_44	dc.w	mml_45-mml_44-2	mml_78	dc.w	mml_79-mml_78-2
	dc.b	'D2-D3,D5'		dc.b	'D1-D3,D6'
mml_45	dc.w	mml_46-mml_45-2	mml_79	dc.w	mml_80-mml_79-2
	dc.b	'D0,D2-D3,D5'		dc.b	'D0-D3,D6'
mml_46	dc.w	mml_47-mml_46-2	mml_80	dc.w	mml_81-mml_80-2
	dc.b	'D1-D3,D5'		dc.b	'D4,D6'
mml_47	dc.w	mml_48-mml_47-2			
	dc.b	'D0-D3,D5'	mml_81	dc.w	mml_82-mml_81-2
mml_48	dc.w	mml_49-mml_48-2		dc.b	'D0,D4,D6'
	dc.b	'D4-D5'	mml_82	dc.w	mml_83-mml_82-2
mml_49	dc.w	mml_50-mml_49-2		dc.b	'D1,D4,D6'
	dc.b	'D0,D4-D5'	mml_83	dc.w	mml_84-mml_83-2
mml_50	dc.w	mml_51-mml_50-2		dc.b	'D0-D1,D4,D6'
	dc.b	'D1,D4-D5'	mml_84	dc.w	mml_85-mml_84-2
				dc.b	'D2,D4,D6'
mml_51	dc.w	mml_52-mml_51-2	mml_85	dc.w	mml_86-mml_85-2
	dc.b	'D0-D1,D4-D5'		dc.b	'D0,D2,D4,D6'
mml_52	dc.w	mml_53-mml_52-2	mml_86	dc.w	mml_87-mml_86-2
	dc.b	'D2,D4-D5'		dc.b	'D1-D2,D4,D6'
mml_53	dc.w	mml_54-mml_53-2	mml_87	dc.w	mml_88-mml_87-2
	dc.b	'D0,D2,D4-D5'		dc.b	'D0-D2,D4,D6'
mml_54	dc.w	mml_55-mml_54-2	mml_88	dc.w	mml_89-mml_88-2
	dc.b	'D1-D2,D4-D5'		dc.b	'D3-D4,D6'
mml_55	dc.w	mml_56-mml_55-2	mml_89	dc.w	mml_90-mml_89-2
	dc.b	'D0-D2,D4-D5'		dc.b	'D0,D3-D4,D6'
mml_56	dc.w	mml_57-mml_56-2	mml_90	dc.w	mml_91-mml_90-2
	dc.b	'D3-D5'		dc.b	'D1,D3-D4,D6'
mml_57	dc.w	mml_58-mml_57-2			
	dc.b	'D0,D3-D5'	mml_91	dc.w	mml_92-mml_91-2
mml_58	dc.w	mml_59-mml_58-2		dc.b	'D0-D1,D3-D4,D6'
	dc.b	'D1,D3-D5'	mml_92	dc.w	mml_93-mml_92-2
mml_59	dc.w	mml_60-mml_59-2		dc.b	'D2-D4,D6'
	dc.b	'D0-D1,D3-D5'	mml_93	dc.w	mml_94-mml_93-2
mml_60	dc.w	mml_61-mml_60-2		dc.b	'D0,D2-D4,D6'
	dc.b	'D2-D5'	mml_94	dc.w	mml_95-mml_94-2
				dc.b	'D1-D4,D6'
mml_61	dc.w	mml_62-mml_61-2	mml_95	dc.w	mml_96-mml_95-2
	dc.b	'D0,D2-D5'		dc.b	'D0-D4,D6'

mml_96	dc.w dc.b	mml_97-mml_96-2 'D5-D6'	mml_130	dc.w dc.b	mml_131-mml_130-2 'D1,D7'
mml_97	dc.w dc.b	mml_98-mml_97-2 'D0,D5-D6'	mml_131	dc.w dc.b	mml_132-mml_131-2 'D0-D1,D7'
mml_98	dc.w dc.b	mml_99-mml_98-2 'D1,D5-D6'	mml_132	dc.w dc.b	mml_133-mml_132-2 'D2,D7'
mml_99	dc.w dc.b	mml_100-mml_99-2 'D0-D1,D5-D6'	mml_133	dc.w dc.b	mml_134-mml_133-2 'D0,D2,D7'
mml_100	dc.w dc.b	mml_101-mml_100-2 'D2,D5-D6'	mml_134	dc.w dc.b	mml_135-mml_134-2 'D1-D2,D7'
mml_101	dc.w dc.b	mml_102-mml_101-2 'D0,D2,D5-D6'	mml_135	dc.w dc.b	mml_136-mml_135-2 'D0-D2,D7'
mml_102	dc.w dc.b	mml_103-mml_102-2 'D1-D2,D5-D6'	mml_136	dc.w dc.b	mml_137-mml_136-2 'D3,D7'
mml_103	dc.w dc.b	mml_104-mml_103-2 'D0-D2,D5-D6'	mml_137	dc.w dc.b	mml_138-mml_137-2 'D0,D3,D7'
mml_104	dc.w dc.b	mml_105-mml_104-2 'D3,D5-D6'	mml_138	dc.w dc.b	mml_139-mml_138-2 'D1,D3,D7'
mml_105	dc.w dc.b	mml_106-mml_105-2 'D0,D3,D5-D6'	mml_139	dc.w dc.b	mml_140-mml_139-2 'D0-D1,D3,D7'
mml_106	dc.w dc.b	mml_107-mml_106-2 'D1,D3,D5-D6'	mml_140	dc.w dc.b	mml_141-mml_140-2 'D2-D3,D7'
mml_107	dc.w dc.b	mml_108-mml_107-2 'D0-D1,D3,D5-D6'	mml_141	dc.w dc.b	mml_142-mml_141-2 'D0,D2-D3,D7'
mml_108	dc.w dc.b	mml_109-mml_108-2 'D2-D3,D5-D6'	mml_142	dc.w dc.b	mml_143-mml_142-2 'D1-D3,D7'
mml_109	dc.w dc.b	mml_110-mml_109-2 'D0,D2-D3,D5-D6'	mml_143	dc.w dc.b	mml_144-mml_143-2 'D0-D3,D7'
mml_110	dc.w dc.b	mml_111-mml_110-2 'D1-D3,D5-D6'	mml_144	dc.w dc.b	mml_145-mml_144-2 'D4,D7'
mml_111	dc.w dc.b	mml_112-mml_111-2 'D0-D3,D5-D6'	mml_145	dc.w dc.b	mml_146-mml_145-2 'D0,D4,D7'
mml_112	dc.w dc.b	mml_113-mml_112-2 'D4-D6'	mml_146	dc.w dc.b	mml_147-mml_146-2 'D1,D4,D7'
mml_113	dc.w dc.b	mml_114-mml_113-2 'D0,D4-D6'	mml_147	dc.w dc.b	mml_148-mml_147-2 'D0-D1,D4,D7'
mml_114	dc.w dc.b	mml_115-mml_114-2 'D1,D4-D6'	mml_148	dc.w dc.b	mml_149-mml_148-2 'D2,D4,D7'
mml_115	dc.w dc.b	mml_116-mml_115-2 'D0-D1,D4-D6'	mml_149	dc.w dc.b	mml_150-mml_149-2 'D0,D2,D4,D7'
mml_116	dc.w dc.b	mml_117-mml_116-2 'D2,D4-D6'	mml_150	dc.w dc.b	mml_151-mml_150-2 'D1-D2,D4,D7'
mml_117	dc.w dc.b	mml_118-mml_117-2 'D0,D2,D4-D6'	mml_151	dc.w dc.b	mml_152-mml_151-2 'D0-D2,D4,D7'
mml_118	dc.w dc.b	mml_119-mml_118-2 'D1-D2,D4-D6'	mml_152	dc.w dc.b	mml_153-mml_152-2 'D3-D4,D7'
mml_119	dc.w dc.b	mml_120-mml_119-2 'D0-D2,D4-D6'	mml_153	dc.w dc.b	mml_154-mml_153-2 'D0,D3-D4,D7'
mml_120	dc.w dc.b	mml_121-mml_120-2 'D3-D6'	mml_154	dc.w dc.b	mml_155-mml_154-2 'D1,D3-D4,D7'
mml_121	dc.w dc.b	mml_122-mml_121-2 'D0,D3-D6'	mml_155	dc.w dc.b	mml_156-mml_155-2 'D0-D1,D3-D4,D7'
mml_122	dc.w dc.b	mml_123-mml_122-2 'D1,D3-D6'	mml_156	dc.w dc.b	mml_157-mml_156-2 'D2-D4,D7'
mml_123	dc.w dc.b	mml_124-mml_123-2 'D0-D1,D3-D6'	mml_157	dc.w dc.b	mml_158-mml_157-2 'D0,D2-D4,D7'
mml_124	dc.w dc.b	mml_125-mml_124-2 'D2-D5-D6'	mml_158	dc.w dc.b	mml_159-mml_158-2 'D1-D4,D7'
mml_125	dc.w dc.b	mml_126-mml_125-2 'D0,D2-D5-D6'	mml_159	dc.w dc.b	mml_160-mml_159-2 'D0-D4,D7'
mml_126	dc.w dc.b	mml_127-mml_126-2 'D1-D6'	mml_160	dc.w dc.b	mml_161-mml_160-2 'D5,D7'
mml_127	dc.w dc.b	mml_128-mml_127-2 'D0-D1-D6'	mml_161	dc.w dc.b	mml_162-mml_161-2 'D0,D5,D7'
mml_128	dc.w dc.b	mml_129-mml_128-2 'D7'	mml_162	dc.w dc.b	mml_163-mml_162-2 'D1,D5,D7'
mml_129	dc.w dc.b	mml_130-mml_129-2 'D0,D7'	mml_163	dc.w	mml_164-mml_163-2

	de.b	'D0-D1,D5,D7'		de.b	'D0,D2,D6-D7'
mml_164	de.w	mml_165-mml_164-2	mml_198	de.w	mml_199-mml_198-2
	de.b	'D2,D5,D7'		de.b	'D1-D2,D6-D7'
mml_165	de.w	mml_166-mml_165-2	mml_199	de.w	mml_200-mml_199-2
	de.b	'D0,D2,D5,D7'		de.b	'D0-D2,D6-D7'
mml_166	de.w	mml_167-mml_166-2	mml_200	de.w	mml_201-mml_200-2
	de.b	'D1-D2,D5,D7'		de.b	'D3,D6-D7'
mml_167	de.w	mml_168-mml_167-2			
	de.b	'D0-D2,D5,D7'	mml_201	de.w	mml_202-mml_201-2
mml_168	de.w	mml_169-mml_168-2		de.b	'D0,D3,D6-D7'
	de.b	'D3,D5,D7'	mml_202	de.w	mml_203-mml_202-2
mml_169	de.w	mml_170-mml_169-2		de.b	'D1,D3,D6-D7'
	de.b	'D0,D3,D5,D7'	mml_203	de.w	mml_204-mml_203-2
mml_170	de.w	mml_171-mml_170-2		de.b	'D0-D1,D3,D6-D7'
	de.b	'D1,D3,D5,D7'	mml_204	de.w	mml_205-mml_204-2
				de.b	'D2-D3,D6-D7'
mml_171	de.w	mml_172-mml_171-2	mml_205	de.w	mml_206-mml_205-2
	de.b	'D0-D1,D3,D5,D7'		de.b	'D0,D2-D3,D6-D7'
mml_172	de.w	mml_173-mml_172-2	mml_206	de.w	mml_207-mml_206-2
	de.b	'D2-D3,D5,D7'		de.b	'D1-D3,D6-D7'
mml_173	de.w	mml_174-mml_173-2	mml_207	de.w	mml_208-mml_207-2
	de.b	'D0,D2-D3,D5,D7'		de.b	'D0-D3,D6-D7'
mml_174	de.w	mml_175-mml_174-2	mml_208	de.w	mml_209-mml_208-2
	de.b	'D1-D3,D5,D7'			
mml_175	de.w	mml_176-mml_175-2		de.b	'D4,D6-D7'
	de.b	'D0-D3,D5,D7'	mml_209	de.w	mml_210-mml_209-2
mml_176	de.w	mml_177-mml_176-2		de.b	'D0,D4,D6-D7'
	de.b	'D4-D5,D7'	mml_210	de.w	mml_211-mml_210-2
mml_177	de.w	mml_178-mml_177-2		de.b	'D1,D4,D6-D7'
	de.b	'D0,D4-D5,D7'			
mml_178	de.w	mml_179-mml_178-2	mml_211	de.w	mml_212-mml_211-2
	de.b	'D1,D4-D5,D7'		de.b	'D0-D1,D4,D6-D7'
mml_179	de.w	mml_180-mml_179-2	mml_212	de.w	mml_213-mml_212-2
	de.b	'D0-D1,D4-D5,D7'		de.b	'D2,D4,D6-D7'
mml_180	de.w	mml_181-mml_180-2	mml_213	de.w	mml_214-mml_213-2
	de.b	'D2,D4-D5,D7'		de.b	'D0,D2,D4,D6-D7'
			mml_214	de.w	mml_215-mml_214-2
mml_181	de.w	mml_182-mml_181-2		de.b	'D1-D2,D4,D6-D7'
	de.b	'D0,D2,D4-D5,D7'	mml_215	de.w	mml_216-mml_215-2
mml_182	de.w	mml_183-mml_182-2		de.b	'D0-D2,D4,D6-D7'
	de.b	'D1-D2,D4-D5,D7'	mml_216	de.w	mml_217-mml_216-2
mml_183	de.w	mml_184-mml_183-2		de.b	'D3-D4,D6-D7'
	de.b	'D0-D2,D4-D5,D7'	mml_217	de.w	mml_218-mml_217-2
mml_184	de.w	mml_185-mml_184-2		de.b	'D0,D3-D4,D6-D7'
	de.b	'D3-D5,D7'	mml_218	de.w	mml_219-mml_218-2
mml_185	de.w	mml_186-mml_185-2		de.b	'D1,D3-D4,D6-D7'
	de.b	'D0,D3-D5,D7'	mml_219	de.w	mml_220-mml_219-2
mml_186	de.w	mml_187-mml_186-2		de.b	'D0-D1,D3-D4,D6-D7'
	de.b	'D1,D3-D5,D7'	mml_220	de.w	mml_221-mml_220-2
mml_187	de.w	mml_188-mml_187-2		de.b	'D2-D4,D6-D7'
	de.b	'D0-D1,D3-D5,D7'			
mml_188	de.w	mml_189-mml_188-2	mml_221	de.w	mml_222-mml_221-2
	de.b	'D2-D5,D7'		de.b	'D0,D2-D4,D6-D7'
mml_189	de.w	mml_190-mml_189-2	mml_222	de.w	mml_223-mml_222-2
	de.b	'D0,D2-D5,D7'		de.b	'D1-D4,D6-D7'
mml_190	de.w	mml_191-mml_190-2	mml_223	de.w	mml_224-mml_223-2
	de.b	'D1-D5,D7'		de.b	'D0-D4,D6-D7'
			mml_224	de.w	mml_225-mml_224-2
mml_191	de.w	mml_192-mml_191-2		de.b	'D5-D7'
	de.b	'D0-D5,D7'	mml_225	de.w	mml_226-mml_225-2
mml_192	de.w	mml_193-mml_192-2		de.b	'D0,D5-D7'
	de.b	'D6-D7'	mml_226	de.w	mml_227-mml_226-2
mml_193	de.w	mml_194-mml_193-2		de.b	'D1,D5-D7'
	de.b	'D0,D6-D7'	mml_227	de.w	mml_228-mml_227-2
mml_194	de.w	mml_195-mml_194-2		de.b	'D0-D1,D5-D7'
	de.b	'D1,D6-D7'	mml_228	de.w	mml_229-mml_228-2
mml_195	de.w	mml_196-mml_195-2		de.b	'D2,D5-D7'
	de.b	'D0-D1,D6-D7'	mml_229	de.w	mml_230-mml_229-2
mml_196	de.w	mml_197-mml_196-2		de.b	'D0,D2,D5-D7'
	de.b	'D2,D6-D7'	mml_230	de.w	mml_231-mml_230-2
mml_197	de.w	mml_198-mml_197-2		de.b	'D1-D2,D5-D7'

```

mml_231 dc.w mml_232-mml_231-2
dc.b 'D0-D2,D5-D7'
mml_232 dc.w mml_233-mml_232-2
dc.b 'D3,D5-D7'
mml_233 dc.w mml_234-mml_233-2
dc.b 'D0,D3,D5-D7'
mml_234 dc.w mml_235-mml_234-2
dc.b 'D1,D3,D5-D7'
mml_235 dc.w mml_236-mml_235-2
dc.b 'D0-D1,D3,D5-D7'
mml_236 dc.w mml_237-mml_236-2
dc.b 'D2-D3,D5-D7'
mml_237 dc.w mml_238-mml_237-2
dc.b 'D0,D2-D3,D5-D7'
mml_238 dc.w mml_239-mml_238-2
dc.b 'D1-D3,D5-D7'
mml_239 dc.w mml_240-mml_239-2
dc.b 'D0-D3,D5-D7'
mml_240 dc.w mml_241-mml_240-2
dc.b 'D4-D7'

```

```

mml_241 dc.w mml_242-mml_241-2
dc.b 'D0,D4-D7'
mml_242 dc.w mml_243-mml_242-2
dc.b 'D1,D4-D7'
mml_243 dc.w mml_244-mml_243-2
dc.b 'D0-D1,D4-D7'
mml_244 dc.w mml_245-mml_244-2
dc.b 'D2,D4-D7'
mml_245 dc.w mml_246-mml_245-2
dc.b 'D0,D2,D4-D7'
mml_246 dc.w mml_247-mml_246-2
dc.b 'D1-D2,D4-D7'
mml_247 dc.w mml_248-mml_247-2
dc.b 'D0-D2,D4-D7'

```

```

mml_248 dc.w mml_249-mml_248-2
dc.b 'D3-D7'
mml_249 dc.w mml_250-mml_249-2
dc.b 'D0,D3-D7'
mml_250 dc.w mml_251-mml_250-2
dc.b 'D1,D3-D7'

```

```

mml_251 dc.w mml_252-mml_251-2
dc.b 'D0-D1,D3-D7'
mml_252 dc.w mml_253-mml_252-2
dc.b 'D2-D7'
mml_253 dc.w mml_254-mml_253-2
dc.b 'D0,D2-D7'
mml_254 dc.w mml_255-mml_254-2
dc.b 'D1-D7'
mml_255 dc.w mml_256-mml_255-2
dc.b 'D0-D7'
mml_256 equ *

```

And that's all there is to it! Hopefully, you have been saving this as you went along, now's the time to assemble it.

I hope you realise that the code we have been typing over the last few issues, with a couple of 'missing' articles, is working code. You can de-code most of the instruction set now - if not all of it. We still have a little way to go though, the separate printing nightmare is still to come :o)

See you next time, where I'll be finishing off here (maybe) or I might even have something totally different. It will still be in assembler though, but, if I get it finished, it might even prove to be useful.

## Corrections

### Correction for an article in Volume 8 Issue 3

#### *David Bunbury writes:*

I expect that you (or someone else) will already have spotted an apparent mistake in the listing on page 43 of the October QT Today but, in case you have not, there is a problem with Line 270, where the GET will advance the file pointer by 6 bytes instead of two.

It works correctly if ql\_bpl is made an INTEGER, which I presume was the intention.

I think that I shall find this program rather useful, although not for Mode 32; I have modified it to work with mode 4 for use with an original "Black Box".

Thank you, David. Probably the easiest correction, to avoid having to rename all

occurrences of the variable ql\_bpl, is to read the value initially into an integer variable then coerce to ql\_bpl:

```
270 GET #3,temp% : ql_bpl = temp% : REMark
bytes per line in QL file
```

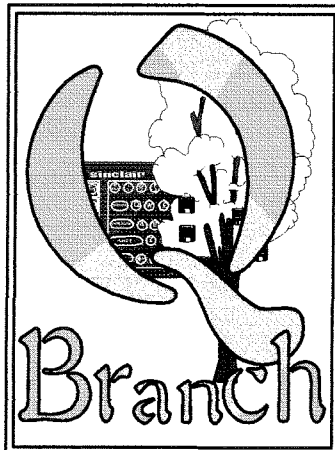
### Correction for an article in Volume 8 Issue 4

There is an error in the listing on page 17 (Calendar program). It's been pointed out we managed to lose a couple of lines from the end of the listing.

Here is the correction:

```
690 DATA 'JANUARY', 'FEBRUARY', 'MARCH',
      'APRIL', 'MAY', 'JUNE', 'JULY',
      'AUGUST', 'SEPTEMBER', 'OCTOBER',
      'NOVEMBER', 'DECEMBER'
700 REMark number of days in months
710 DATA 31, 28, 31, 30, 31, 30, 31,
      31, 30, 31, 30, 31
```

Dilwyn Jones



**Text 87**  
 £ 79.00  
 Typset 94 £ 29.00  
 Fountext 94 £ 39.00  
 2488 drivers £ 29.00  
 Epson ESC/P2 drivers  
 £ 26.00  
 Text 87  
 High Colour  
 Patch  
 Available Now  
**£9.00**

**PROGRAMMING**

QD 2003	£ 49.00
QD + QBasic	£ 63.00
QD + Qliberator + QBasic	£ 104.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 32.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 34.00
QMenu	£ 16.00

**Many Happy ENTER's**

This year is the twentieth birthday of the QL. Why not celebrate by treating yourself to all these colour upgrades. The Text 87 High Colour patch is released now - but don't forget it will only work with QPC2/QXL. Patches for other systems may follow but we need to know how many people are interested. Contact us if you are.

There are still a few keyboard membranes available to get that old black box up and running again.

For those of you who need a basic printer for your QL we can get the EPSON LQ-300+ which is a 24 pin Dot Matrix model with ESC/P2 emulation. Price £135.00 plus shipping. Call for availability.

**Just Released!**

**High Colour Drivers for Gold / Super Gold Card SMSQ/E**  
 (Aurora only - Send old master disk as proof of purchase)

**£25.00**

**QL Keyboard Membranes**

**Available Now !**

**£ 17.50 ea + P&P**

These programs are free upgrades :

- Agenda
- QMake
- Wined
- FiFi II
- QPAC 1
- QPAC 2
- QSup

All versions of

SMSQ/E(Gold Card/QXL/Q40

(Just send 75p for the disk and 80p postage + master as proof of ownership)

**UTILITIES**

FiFi 2	£ 21.00
QSup	£ 30.00
QSpread 2003	£ 51.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 22.00
QPAC 2	£ 42.00
QTYP 2	£ 31.00
QLQ	£ 30.00

**These Programs are paid Upgrades:**

QSpread 2001 to 2003	£ 10.50
QD 98 to 2003	£ 10.50
QPC2	-
update from v3.xx	Free
update from v2	£ 13.90
update from v1	£ 34.00

**Return Master Disk With Order**



Tel +44 (0) 1273-386030 fax +44 (0) 1273-430501

Mobile +44 (0) 7836-745501

email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

# Q Branch

Feeling out on a limb?

Reach out for Q Branch.

Suppliers of Quality QDOS/SMSQ products  
Hardware and Software.

20 LOCKS HILL, PORTSLADE,  
E. SUSSEX. BN41 2LB. UK.

## ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

## Paragraph

The ProWesS word processor

**P** Demo version £ 1.50 + postage  
Full Registered version £ 18.00

Version 2.03 available now !

## Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

<b>QXL II</b>	£ 100.00*
Recycled superHermes	£ 65.00 *
R'cycl'd SuperGold Card	£ 110.00*
Recycled Gold Card	£ 45.00 *
Recycled Aurora	£ 70.00**
Qubide	£ 50.00*
Qplane	£ 5.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

\* when available.

Call for details

Qubide upgrades to version 2.01 £ 8.00

Gold Card / Atari / QXL Version  
Version 3.04 out now!

Now Only £ 32.00

### QPC 2 v3.11 - full colour version!

Special offer !  
Get Cueshell for only £15.00 with any copy of SMSQ/E  
or QPC 2 upgrade !

## SMSQ/E

Aurora High Colour  
version available now

upgrade £ 21.00

Now Only £ 65.00

## Q Branch Programs

The Knight Safe 3	£ 35.00	Q - Route v1.08C	£ 25.00
upgrades from previous versions	£ 5.00	Route finding programme	
Q - Count	£ 25.00	The Fractal Collection	£ 35.00
Pointer driven home accounting			

Post and Packing is included with Software Prices  
Unless Otherwise Stated.



We can accept payment by Visa, Mastercard and Switch. We also accept  
cheques drawn on a UK Bank in Sterling.

You can also pay directly into our bank in Euros.

Barclays Bank Acc# 79016888 sort code :20-71-02 Name :QBranch)



Q Branch

# Windows in the Pointer Environment using SETW

George Gwilt

SETW is a program providing an easy way of defining windows for the Pointer Environment for both the new and old versions of WMAN. The program will run on all QL type machines from an AH rom with Trump Card to a Q60 or QPC.

## General

One of the more annoying stumbling blocks to programming for the Pointer Environment is the production of the definition of the windows. A window definition for PE goes far beyond the simple size, position, ink and paper colours of SCR/CON.

A full PE window definition contains a range of sizes for the main window and within each of these contains information about loose menu items and sub windows and their contents. The sub windows must be one of two types, either information, which is simply a defined area within the main window, or application, which can contain a menu and allows actions to be chosen from within it.

The contents of sub windows can be text, sprites or blobs with patterns.

The original aids to a programmer defining a window are the Qmac macros for use with assembler language and the extension keywords of QPTR for use with SuperBASIC. In both cases the sizes, positions and origins of all the items have to be given directly as numbers. You can, of course, use squared paper to help in defining these numbers, but until you see the window displayed on the screen by the PE software you will not really know how it is going to look, or indeed whether it works at all.

Unfortunately, for a proper display all the constituent parts of the window must be compatible. If an object is defined to be outside its allotted area, even by so much as a pixel, the window will not appear. The PE software then collapses with the message "out of range".

Clearly this is very frustrating for the programmer.

This problem was addressed in both TurboPTR and CPTR by the programs SETF (in TurboPTR) and SETZ (in CPTR). These programs allow a programmer to specify the contents of a set of windows and to set the sizes and positions by manipulation on the screen. Thus the window is

built up under the programmer's eyes as he proceeds. Also each element in the window is forced to be within acceptable limits. For example, text cannot be set to be partially outside the window or sub window which must contain it.

SETF and SETZ thus allow a visual development of the window and its contents. When the operation is complete all the windows are shown in turn as they will appear when used by the program for which they are designed.

Both SETF and SETZ and their use are described in two articles in QL Today. The first, on page 39 Vol 7 Issue 4, contains in an Appendix a note of the operations of SETF as seen by a programmer. In the second, on page 12 of Vol 7 Issue 6, John Sadler shows in detail how SETZ can be used to produce a working program.

## SETW

The new version of WMAN, which appeared first in SMSQ/E v3.00 and which will presumably in due course be made available also to other QL operating systems, contains such a range of extensions in the definitions of colours and sprites that an update of SETF and SETZ became essential.

So extensive are the changes to both colours and sprites that I had thought first of producing two new versions for both SETF and SETZ for the new WMAN, leaving the original programs for use with the old WMAN. I judged this to be too confusing, so instead I have produced just one program, SETW, combining all the four features.

The new program SETW is thus a combination of the two previous programs SETF and SETZ. It produces output suitable for both TurboPTR and CPTR. SETW also tailors the output to the type of WMAN present when it is run.

## Output

SETW produces two files as output with names ending in `_WDA` and `_z`. They are first a set of DATA lines incorporating the complete window definition in a form suitable for TurboPTR, and second a set of C instructions suitable for CPTR. (See John Sadler's article mentioned above.)

Although the `_WDA` file is not needed to produce a C program via CPTR, it is nevertheless useful in that the program "SEEWIN" can be used to display the window. This is particularly useful where colours are set to the system palette defaults. In this case the new feature of "SEEWIN" allowing specification of a particular system palette can be used to view the window with different colours for the defaults.

Also the `_z` file of C instructions is not used by TurboPTR, but is effectively a useful record of the values of the various items in the window definition.

## Colours and Sprites

Since the operation of SETW is essentially the same as that for both SETF and SETZ I will not describe it in detail here. However the new colours and sprites required some revision of the process and these advances are dealt with here. When a colour is to be chosen SETF/Z shows one block of colour along with a description. The full range of QL colours available can be seen, one at a time, by using the up and down arrow keys.

In place of this SETW shows a window containing 16 colours from the full range. The colour currently selected is indicated by its description being white and not green like the others. Also there is a scroll bar to the right of the window which shows the proportion of the full range shown in the window and its position. Selecting a colour is achieved as before by pressing the up and down arrows. In addition if ALT is pressed too, the display alters by a window depth.

The same process, incidentally, applies to the selection of an item from the user's list of text items.

As with the colours, SETF/Z presented the user with just one sprite and its description. Movement through the total list of available sprites was done as for colours by using the up and down keys.

SETW shows as many sprites as will fit into the display window and allows selection by use of the up and down keys. However, in this case the ALT key has no effect.

## New and Old WMAN

### Colours

The old WMAN only caters for the byte range of QL mode 4/8 colours. The new WMAN has seven types of definition of colour which can be used in the window definition. These are described in the article by Wolfgang Lernerz on page 47 of Vol 8 Issue 2 of QL Today. They are represented by SETW as:

Default	(System Palette)
COLOUR_QL	
Gray Scale	
COLOUR_PAL	(Palette)
COLOUR_PAL & Stippl	(Palette & Stipple)
COLOUR_24	(RGB)
Border	

When a colour is to be chosen and SETW detects the presence of the new WMAN, it displays first a window giving the choices shown above. Unless Default is chosen, a new window appears. In the case of COLOUR\_QL, Gray Scale and COLOUR\_PAL the window now contains the first 16 colours in the range chosen. For COLOUR\_QL the range is exactly the same as for the old SETF/Z. For Gray Scale the range covers the 63 different shades. COLOUR\_PAL has a range of 256.

If one of the remaining colour types is selected, an outside program is called. Currently these three programs merely ask for the values of the three parameters in each type. No colour is displayed in advance though when the choice is made the result is shown and the option is available to try the selection again.

The "Default" option sets a particular entry to the system palette depending on what the colour is to be used for. The actual defaults are shown in an Appendix below.

## Sprites

Under the new WMAN the number of system sprites has been increased from 8 to 38. Many of these additional sprites, such as "F2" are already available in C68 and are incorporated into the file `cptr_o` used by CPTR. To avoid duplication, a new, shorter, file called `cptr2_o` has been produced. The sprites displayed for choice by SETW are, for both old and new WMAN, first the system sprites (8 or 38 in number), followed by the sprites selected by the user and finally the sprites included in `cptr_o` or `cptr2_o` as appropriate.

With the new WMAN, any of the sprites can be used for a loose item, a window sprite for the main window or a subsidiary application window and finally for an item in an information window. There is one proviso regarding TurboPTR. A system sprite used as a loose item must be one of the first ten loose items for the old WMAN and 28 for the new WMAN.

Under the old WMAN there are more restrictions. The use of system sprites is restricted to loose items. They can't be used as window sprites, or items in an information window. Furthermore, in addition to being restricted to the first ten loose items in TurboPTR, only the move and resize sprites can be used in CPTR, any other system sprites selected and used in TurboPTR are forced to the default arrow sprite.

## Appendix Default Colours

One of the colour options in the new WMAN is the system palette. The value of the "colour" in this case is the number of an entry in the palette. The actual colour in a program used will be the value of the particular entry in the system palette set to the program at run time.

In order to allow some uniformity of practice the various entries in a system palette are assigned specific names. The subset of the 57 entries currently available used as defaults by SETW are given below:

Main Window Border	\$200
Main Window Paper	\$201
Loose Item Border	\$207

Available	- Paper	\$208
Available	- Ink	\$209
Selected	- Paper	\$20A
Selected	- Ink	\$20B
Unavailable	- Paper	\$20C
Unavailable	- Ink	\$20D
Information Window	Border	\$20E
Information Window	Paper	\$20F
Information Window	Ink	\$210
Application Window	Border	\$216
Application Window	Paper	\$217
Menu Item	Border	\$21A
Available	- Paper	\$21B
Available	- Ink	\$21C
Selected	- Paper	\$21D
Selected	- Ink	\$21E
Unavailable	- Paper	\$21F
Unavailable	- Ink	\$220
Scroll Bar		\$221
Scroll Section		\$222
Scroll Arrows		\$223

# Letter-Box

*We received an interesting email from Simon Goodwin in response to some of the articles in the last issue. Here is a slightly edited version, with my responses in italics where I thought appropriate - Editor.*

### Launchpad Cover Disk

Pity Launchpad is dependent on the PE, and therefore no use to me. However the generic (non-PE) parts of QL Today are still much appreciated.

*We do try to cater for both PE and non-PE users, but the majority of material received for publication does seem to be PE orientated. We would, however, happily publish more non-PE material if more were to be sent to us! What do you the readers think about this - do you think we have the balance about right, or do you wish to see more of one or the other?*

### QL Printers

You might usefully add some Brother printers to your list. I use an HL-1050 laser (£100 second hand about three years ago) and there are many similar models (HL-1030, 1040, 1070, 1240, 1250, 1270N).

Its an uncommonly cross-platform compatible printer - like the HL-1070 but with only one paper

input bin (the 1070 has two). The fast RISC processor is easily expandable with 72 pin SIMM memory (from 4 to 32 Mb) but unless you use big graphics at top resolution the internal 4 Mb will probably be enough.

This printer has automatic HP-PCL and Epson ESC-P sensing, so it works with many PCL and Epson drivers, selecting the right emulation on the base of the first control codes it gets. So it works with the DIY Toolkit DeskJet drivers, Gold Card Epson drivers, and many others.

It also automatically services parallel and USB ports by initially signing on as a USB printer and then sending a 'disconnect' message to the USB host if the parallel input port becomes active, printing that data then 'reconnecting' to the USB host after that job ends.

This means it shares well between old and new computers with Centronics and USB printer interfaces respectively, without need of a switch box. I use the parallel end on my Sam, Amiga Qdos and QL (via serial Centronics adapter) systems, with a switch box, and can drive the USB from UQLX (or Linux in general).

A good source of information about which printers have brains (and therefore print text as fast as they can shift the paper, unlike the Win-doze GDI ones that only accept bitmaps) is the guide at <http://www.linuxprinting.org/>

I've found Roy Wood's complaints in QL Today misleading - there's still no shortage of printers compatible with Qdos, as long as you know what question to ask. I hope this information will help others.

## WHEN Variable

WHEN variable is rather more robust on MG than on JS versions of Qdos. As it's primarily a debugging feature it's potentially useful on any system when you're not sure what's clobbering a variable. The HOW\_COME PROCedure in Turbo Toolkit Demos and DIY Toolkit Volume B is handy for working out WHAT triggered the WHEN exception.

It is in all the Argos (Thor XVI OS) ROMs, with much the same level of functionality as on MG ROMs (surprise, surprise). It is not implemented in Turbo and not likely to be since it would involve inserting a conditional test on each write access to scalar variables, making every program that might use it substantially slower. That said, perhaps it could be done setting the bottom bit (0) in the Vector Table entry for the variables you wanted to watch, and trapping the exception, but this would trap reads as well as writes and wouldn't work on a 68020 as that allows integer and floating point values to be at odd addresses (not that Turbo will ever put them there normally, as misaligned access is much slower).

## Tiptoeing Through Turbo

George Gwilt

I am always glad to hear of better ways of doing things, especially from Simon Goodwin who unfailingly hits all nails on the head. So when I learnt that the way to go through Turbo's threaded code was to use the "move, jmp" link between each template, I immediately tried it out on two very short programs with the following results.

### Program Number 1

```
100 PRINT "noom"
110 STOP
```

I compiled this with options - 1 window structured include line numbers brief

I discovered that the "move, jmp" code was at \$322 from the start of the compiled program and set a breakpoint there. All agog I pressed 'g' to set Qmon off! What happened? This happened:

Press Nr	Result Seen	Action Taken
1	No Program yet	"g" pressed
2	No Program yet	"g" pressed

## Stepping through TURBO Code

George Gwilt does not appear to be aware of how to step through Turbo threaded code. The trick is to compile with size optimisation - this means that each template ends with a JMP instruction to shared code to fetch the next thread address, rather than have a copy of the six-byte threading routine at the end of each template. So a breakpoint on the generic instance of this:

```
move.w    (a5)+, d0
jmp       0(a6, d0.w)
```

which appears in the first template (ENTRY\_ROUTINE) and at the same offset at the start of every compiled program, picks up each code-generated instruction (CONSTANT\_FLOAT, MULTIPLY, ROM\_FN etc) without the need to step through the code that implements the templates. Further reading: Turbo docs and open source!

**We brought this to George's attention and he has responded to this in the following article.**

3	Program started - window drawn	"g" pressed
4 to 12	No change	"g" pressed
13	"noom" printed	"g" pressed
14	No change	"g" pressed
15	No change	"g" pressed
16	Job gone	

### Program Number 2

```
100 PRINT "noom"
105 SUSPEND_TASK 200
110 STOP
```

The "move, jmp" code was indeed again at \$322 from the program's start. The results were the same as for the previous program until after press number 15 when:

Press Nr	Result Seen	Action Taken
16	No change	"g" pressed
17	No change	"g" pressed
18	Program halted for 200 ticks	"g" pressed
19	No change	"g" pressed
20	Job gone	

It seems that this technique allows you to go through Turbo compiled programs skipping the templates. I have never yet wanted to do that but perhaps the time will come. The problem that I had with Turbo was to see what exactly a particular template was doing. In my example the

template was that for DIMN. I think I would rather stay with the second method I suggested than calculate how many g's I would have to press before I arrived at the DIMN template in the program containing it.

In the examples above there were 16 to 20 templates. What were they all?

It occurred to me that perhaps Simon expected me to cut down on the number of threaded code templates by using the REM +, - facility to turn on and off fast code which does not use templates. When I tried to isolate the PRINT command in the above examples by this method, I still found that I had to step through several sets of threaded code (obviously not all of these are turned off by the REMs) before I arrived at the PRINT com-

mand. I was only able to determine that I had arrived at that command by examining the output. Of course by the time I had seen that "noom" was printed it was too late. I had to restart the whole process, this time counting carefully through the number of times I had to press g.

However, I will say that Simon's method combined with my MAGENTA% method would have saved a few key presses, since I could have jumped from the start of MAGENTA% to the code loading the next template (which I wanted to investigate).

Unfortunately I have abandoned the MAGENTA% method.

I have, though, learnt the magic number \$322 which I will bear in mind. Thanks Simon.

---

## The new GD2 sprite system

*Marcel Kilgus*

With the arrival of SMSQ/E v3 the sprite capabilities of the PE have been greatly enhanced. Applications like the new QD already use high colour sprites and some people have wondered how those were done and how one can do something like that. Well, the first step to get a nice sprite is to write Phoebus Dokos and tell him what you want ;—)

Fortunately he did volunteer to do the new JMS sprites because neither me nor Bernd Reinhardt, who with my help did most of the work of converting the JMS applications to high colour, are really that good graphics designers. If you don't have the privilege of a personal graphics guy you probably have to fetch your mouse and draw the sprites yourself.

Unfortunately this step is best done on a PC because even the simplest applications there are far superior to anything SMSQ/E currently offers. QD's sprites for example were done in Adobe Photoshop.

Alternatively one can search for a motive on the internet or go through various clipart collections. In any case the goal is in the end to have a graphics file in the PNG file format.

PNG stands for "portable network graphics" and the format got really popular due to patent troubles with the established GIF files. Most PC applications can read and write it nowadays. The

format offers lossless compression of the graphics data in various colour depths and last but not least something called an "alpha channel".

I should probably explain first what this is. QL sprites for example have a "mask". The mask says what pixel of the rectangular graphics data should be drawn on screen and which should be left out. This way one can draw graphics objects that are not rectangular, like the standard arrow sprite. This is called "hard transparency", which simply means that a pixel is either visible or not.

Now an alpha channel is like a mask but it offers more: for every pixel in the graphics data there is one value in the alpha channel which can usually range from 0 to 255. 0 means "pixel is not visible" and 255 means "pixel is visible", just like in a mask. The interesting part happens if you chose values other than 0 or 255. In this case the system mixes the background pixel with the sprite pixel and thus creates a "soft transparency". "10" for example means "mostly background colour with a shade of the sprite data". The higher the value, the more visible the sprite pixel gets.

Perhaps you have seen a Windows pointer sprite that drops a soft shadow on the desktop below. In the graphics data this shadow is really just black. Only the alpha channel creates the effect of a soft blending between the shadow and the background.

As I like eye candy effects I just had to implement this into the SMSQ/E graphics drivers and since version 3.00 it is available to the public.

A complete pack with replacement system sprites (like the arrow, window move etc.) that use the alpha channel for shadows or

# TF Services

## Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets. Can be used with lights/hifi/monitors—ie a QL monitor can be used as a switch control.

Cost **£24**

## superHermes

A major hardware upgrade for the QL. All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/F // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software) ..... **£90** (£92/£93)  
 IBM AT UK layout Keyboard..... **£11** (£13/£15)  
 Serial mouse..... **£8** (£8.50/£9)  
 Capslock/scrolllock LED ..... **£1** (£1.50/£1.50)  
 Keyboard or mouse lead ..... **£3** (£3.50/£3.50)  
 High speed serial (scr3) lead..... **£4** (£4.50/£4.50)

Hermes available for **£25** (£26/£27) Working ser1/2 and independent input, debounced keyboard.

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.  
 Cost (incl keyboard lead) ..... **£53** (£54/£55)

## QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

**£27** incl 6 month guarantee

## Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send **£3** (+£5 for manual if reqd).  
 Send disk plus SAE or two IRCs

MKL..£40 (£41/£43) MKII..£65 (£66/£67)

**MINERVA RTC (MKII) + battery for 256 bytes ram.**  
**CRASHPROOF clock & I<sup>2</sup>C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.**

## QL RomDisq

Up to 8 mbyte of flash memory for the QL. A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second. Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£40/£41)  
 4mbytes RomDisq..... **£65** (£66/£67)  
 8 mbytes RomDisq..... **£98** (£99/£100)  
 Aurora adaptor..... **£3** (£3.50/£4)

## MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost ..... **£34** (£35/£36)

## I2C INTERFACES

Connects to Minerva MKII and any Philips I<sup>2</sup>C bus

**Power Driver Interface** 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors)..... **£40** (£43/£44)  
 4 amp total (for motors etc)..... **£45** (£48/£50)

**Relays** (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£29)

**Parallel Interface** Gives 16 input/output lines. Can be used wherever logic signals are required..... **£25** (£27/£28)

**Analogue Interface** Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting..... **£30** (£31/£32)

**Temp probe** (-40°C to +125°C)..... **£10** (£10.50/£11)

**Connector for four temp probes**..... **£10** (£10.50/£11)

**Data sheets**..... **£2** (£2.50/£3)

**Control software & manual (for all I/F)**..... **£2** (£2.50/£3)

## QL SPARES

Keyboard membrane ..... **no longer on sale**

1377 PAL ..... **£3** (£3.50/£4)

Circuit diagrams ..... **£3** (£3.50/£4)

68008 cpu or 8049 IPC ..... **£8** (£8.50/£9)

8301/8302 or JM ROM or serial lead ..... **£10** (£10.50/£11)

Power supply (sea mail overseas)..... **£12** (£19/£23)

Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable) Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address/postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details

22 Feb 03

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: +44 (0) 1442-828254

Fax/BBS: +44 (0) 1442-828255

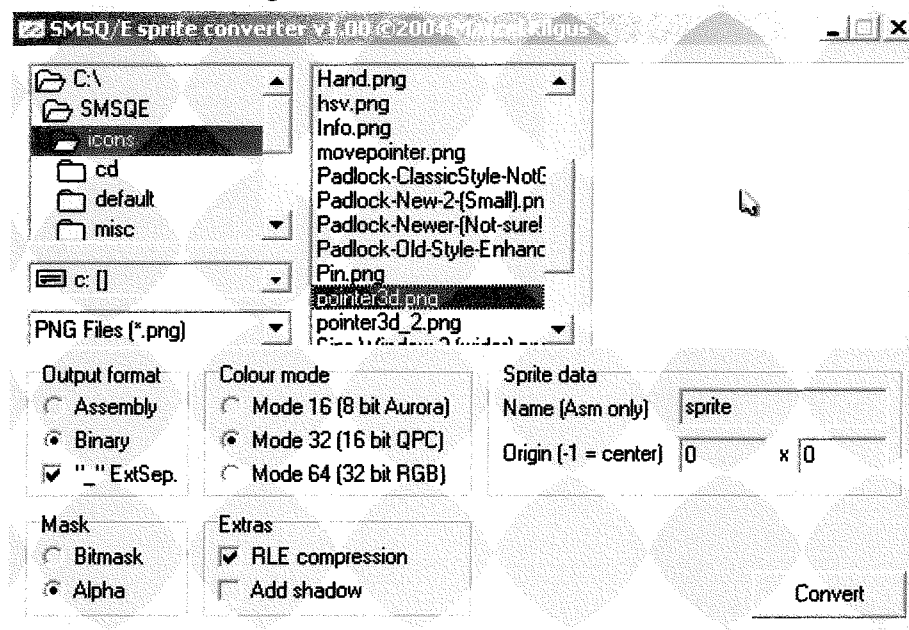
tony@firshman.co.uk

http://www.firshman.co.uk

see-through transparency is available from my page at <http://www.kilgus.net/smsqe/>

Look in the "GD2 high colour" section. The new QD toolbar sprites also use transparency if you look closely.

Now, back to the PNG files. In order to create an SMSQ/E sprite out of the PNG files you need another tool called "SMSQ/E sprite converter" which is also available from my site. It is a Windows tool, simply because it was much faster to program this way. The basic usage of the application should be quite obvious: navigate to the PNG file you want to convert, ensure through the preview window that you have the right file, hit "Convert" and that's it. There are however a few more options that might need some explanation:



**Output format:** the converter can output assembly source code or a binary file that is directly usable as a sprite. You can choose between those two here. Also for convenience you can say whether the result should have an underscore as a file extension separator or a dot (i.e. "\_spr" vs ".spr").

**Colour mode:** this sets in what format the sprite should be saved. You can choose between mode 16, mode 32 or mode 64. Mode 16 is basically the 256 colour mode of the Aurora graphics card. The advantage is that in this mode the resulting file is small but the downside is that the graphics quality can suffer severely. Mode 32 is the 16 bit graphics mode of QPC/QL and is often the best choice. Mode 64 is the 32 bit graphics mode that doesn't improve image quality but heavily increases the file size.

**Sprite data:** Here one can give the sprite a name which will be used for the labels in the assembly

output. If you chose binary output in the format settings the name is ignored. Also you can say what origin the sprite should have. Set it to -1 to simply center the sprite origin.

**Mask:** This is the magic switch that decides whether the old hard transparencies should be used ("mask") or if you want a softer blending into the background ("alpha"). A sprite with a mask is a bit quicker to draw compared to an alpha sprite but often also bigger in size.

**Extras:** Here you can enable the RLE (run length encoding) compression which can reduce the size of the resulting sprite file. This is also a new thing implemented by me into the SMSQ/E graphics core because high colour sprites can be pretty huge in memory consumption. The compression is not highly efficient with every kind of

sprite, by far not as good as that used for PNG files, but on the other hand it is very fast and pretty easy to implement. The downside to using RLE compression is only that at the time the sprite is drawn the system needs a bit more memory and time to first decompress the sprite data. But this should hardly be noticeable. Also there is a switch that when enabled automatically computes a soft shadow to the given graphics.

This is mostly useful if you want to create pointer sprites with a shadow like it is known from Windows.

Now, with this knowledge you should be able to create your first high colour sprites. But how can the files be used? Well, this is probably worth an article of its own. But in principle you can use the sprites in the same way like traditional QL sprites. EasyMenu for example, although written a long time ago, can handle high colour sprites just fine. What my converter can't directly create are hybrid sprites that work both in QL and in high colour mode. But those are not that difficult to do. Basically the high colour sprite just has to be put right after the QL sprite and after that at file position 20 one has to write the length of the QL sprite file minus 20 (as a long word). I think this might be a good exercise for my dear readers to do (if at this time I still have some). Another thing are sprites that change depending on whether the menu item is selected or whether the mouse





is currently hovering over the item. Those can be created using Wolfgang Lerner's BMP2SPRT application, the "competition" to my sprite converter. It is a very good looking and free SMSQ/E application that is certainly worth a look. It is available from

<http://www.scp-paulet-lernerz.com/14mijkl24/Wolf/Download>

Well, I hope this article gave at least a little insight into the new high colour world and I would love to see new applications that make use of the new possibilities.

---

## Launchpad Update

John Perry

A few issues ago I wrote about the early version of Launchpad I'd been using for a very short while, and how promising I found it despite the inevitable bugs and shortcomings of pre-release software. Features were missing, not fully working and poorly documented; despite that I found it surprisingly useable.

So now that I have been using the full release version for a while, I thought it time I updated my review. I will be writing about version 0.96, the most recent copy I have been given. I'm pleased to say that things have only got better! Some new features have been added as a result of pleas by people like me. Some features have been enhanced. All the little accessory programs are now included and working. There's a manual. And above all the program is now much more bug free than the original version I reviewed. Even the Help system now works!

First of all, if you don't like the pointer environment, you won't want to use Launchpad. It is pointer driven by its very nature, it simply won't work without pointer environment. You can CTRL C to SuperBASIC and type in commands from there in much the same way as you can go to DOS from Windows and type in commands from there but that's as close as you get. So if you aren't a pointer environment user, stop reading now (unless

you want to know what you are missing).

Don't have a mouse? You can use Launchpad with cursor keys and space bar and enter to select items from menus and so on but it defeats the object. I won't go into the theory of Launchpad too much, my original review covers that, enough to say that it puts a graphical user interface onto QDOS or SMSQ. You can set up icons and now menus to start or "launch" programs. You are given a few little games and other accessory programs and a substantial manual to get it all going, not that you really need it too much. The manual is a set of Quill DOC files (also supplied as plain text files which also form the Help files in the program). You also get a couple of printed sheets as a kind of quick-start guide to help you install and start the program.

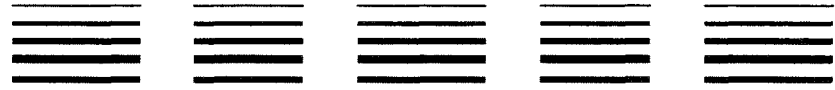
Things get off to a really bad start - you get the program on two floppy disks. Except it won't run from the floppy disks! I contacted the author about this and he claims it's deliberate. You have to read the two sheets of printed instructions and configure the program correctly to get it to work. Seems the idea is to force you to read the instructions before you can (ab)use the program. It makes sense in some ways I suppose, unless you install it correctly, you might not get it working at all or not get the best results from it. To quote the author, if a

phone call goes along the lines of "I've tried running the disks, it doesn't work", he knows the user hasn't read the instructions. I can see that this would potentially be a good thing, but it does run the risk of annoying customers.

In actual fact, the configuration process is quite simple. You simply copy the program to a blank disk or to a directory on your hard disk and then run the CONFIG program to specify where it's been installed. If you create a directory called WIN1\_LAUNCHPAD\_ and put it in there, then run config and enter that directory name and one or two other little details and that's it. Apart from editing the boot program if you use the supplied boot program. All easy enough, but think how the QL differs from the PC. Installing a program on a PC under Windows usually consists of running an installation program, often a little 'wizard' program which guides you through each step. While the last thing I want is turn my QL into a Windows machine, I do think there are areas in which both Windows and the QL could learn from each other, automatic or semi-automatic program installation is an area where the QL could learn from Windows. However, Launchpad installation is pretty much about as simple as most QL software goes and this should not be seen as too much of a criticism.

Having installed it, I ran it and BANG. A configuration error occurred. Oh dear, here we go, I thought. Then it occurred to me that I'd installed the new

# QUANTA



## Independent QL Users Group

World-wide Membership is by subscription only,  
offering the following benefits:

Bimonthly Newsletter – up to 40 pages

Massive Software Library – All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: MasterCard

**\*Now in our Twenty First Year\***

Further details from the Membership Secretary

**John Gilpin, 181, Urmston Lane  
Stretford, Manchester, M32 9EH**

**Tel. +44 (0) 161 865 2872**

or

**Visit the Quanta Web Site**

***<http://www.quanta.uni.cc>***

***E-mail: [quanta\\_membership @ uk2. net](mailto:quanta_membership@uk2.net)***

version over the old version rather than deleting the old version first. And the old version was a pre-release version. So I tried again with a clean installation and it worked first time. Such is the joy of being a software reviewer. I later learned that new facilities have been added to the program which has caused a significant change to the configuration and setup files so the current version cannot find some information from files created on pre-release versions, although it can cope with configuration files from more recent versions if you upgrade. Basically, if you paid for it and later update it should work with older configuration files by substituting sensible default values for any missing items in these files.

When you start the program what you get is basically a blank white screen. Nothing is set up when you first start it, you have to set up your own program icons or menus.

And this is one of the biggest differences between this and the version reviewed. Menus. The original version could only launch programs via icons defined on the desktop. See the screen dumps in the July/August 2003 issue of QL Today to see the program icons. I originally wrote that I'd like to see a facility akin to the Windows START menu, where programs are grouped into lists and you select which program to launch from a menu. Launchpad now has such a facility, called LAUNCH or QUICK-LAUNCH (surprise surprise!). In some ways, it's like a Windows start menu (a bit less comprehensive perhaps) with a touch of the QL Qascade program about it. A little confusingly, the menu originally called LAUNCH has now been given this new

facility, while the old LAUNCH menu is now called UTILITY, which makes sense once you get used to it, as it perhaps better describes the facilities offered.

The Launch menu starts as just one empty menu. Here you can add an item to start a program (or anything which can be started by File Info 2 if you use that), or add a sub-menu if you wish to group programs of the same type together. Sub-menus can contain sub-sub-menus and so on, it can get pretty complex if you try hard! For example, you could create a menu called GAMES to put all your game programs onto. You can create a Word Processing menu, databases menu and so on to your heart's content. Menu names can be fairly long, nearly as long as QL filenames, though certain characters are not allowed (spaces and underlines seem to be allowed so you can set up pretty meaningful names) since they are used as internal flags, it probably means that the few restricted characters like arrow symbols are used internally to signify menu names or filenames or whatever.

Visually, there is not a lot to the Quicklaunch menu. Figure 1 shows what they look like. Basically, a white rectangle with the usual move and ESC icons at the top, a list of programs or menus in the middle and three commands at the bottom to add a new program to this menu, add a new sub-menu and a little ← icon to let you go back to the previous menu. A hit (left mouse click) on an item in this menu enters the sub-menu if the item was a menu name, or

starts the program if it was a program item (or tries to start the item via File Info 2 if relevant and the item was not an executable program). Things get a little bit more complicated if you do (right click) on one of the items in the menu. A new menu appears, with options to alter the entry, delete it, launch it (with various further options!), pick that program if it's already running, or sort the menu itself into alphabetical order. Personally, I'd like to have seen items sort themselves into order automatically as they are created, but at least this way you have the option of sorting or not I suppose.

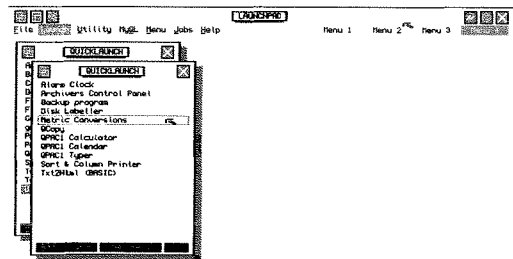


Figure 1 - The Quicklaunch Menu

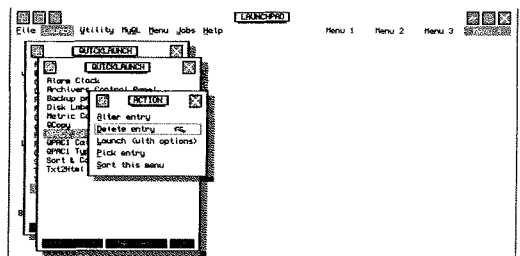


Figure 2 - The Quicklaunch Menu's Action Menu

All this makes the Quicklaunch system seem too complex. In fact, a lot of thought has gone into this and it only gets as complex as you make it. Setting up the menus initially is a bit time consuming but pretty straightforward - you specify the filename (you can either

type it in or select it from a file selection menu), the name which appears in the menu itself, the job name (so that Launchpad can PICK the program, though in most cases it automatically extracts this from the program by itself, you can change this if you know you are dealing with a program which changes its own name as it runs) and a 'command line' to feed to the program as it starts. For example, this could be used to specify which file a program is to load as it starts up, or to tell some programs where to find their own files. Like most things in Launchpad, it can get as complex as you want it to get, simple useage generally involves least effort setting it up, complex useage takes more time and effort to set up.

Difficult programs are not so easy to set up in the Quick-launch menus. Desktop icons can have DEV, SUB, memory protection etc etc set up, Quicklaunch items can't. So programs like Quill which will need memory restrictions and possibly DEV or SUB settings still have to be set up on the desktop icons and can't really be put in the Quicklaunch system. Not a big restriction (how many people still use Quill, Archive etc?)

The hardest part with Quick-launch is setting it up in the first place. Once set up, it's a dream. It seems to handle all the programs you could need and more (I don't know if there is a limit on the number of programs or menus it can handle, it seems to have coped with everything I've thrown at it so far!). It doesn't 'know' about programs, you manually select filenames or pick them from menus for example, but once set up, you just go to the menu you want and left click on the

name and off it goes. This is going to simplify my QLing no end! My only gripe is it should have been in Launchpad from day one!

A major improvement has been made in terms of handling display resolutions and colour depths. In the early versions, you specified resolution and colour mode and it made you use those settings. If Launchpad decided you wanted to use VGA resolution in high colour mode, no matter what settings you specified before it loaded, it insisted on trying to change your computer to what it thought you last set in Launchpad. Now, it can still do this (see it as setting your preferred settings) but you can also set it to just start up in whatever mode you have your computer set to. This is not always ideal, but it seems to manage and I'm sure a lot of users will find this a much better way of doing things.

The Help system is now implemented, although not quite in the way I'd expected. It brings up a menu, letting you read the relevant instruction file via its own text file viewer - basically, plain text versions of the manuals. The help is not context-sensitive as it is in a Windows system, for example.

Desktop icons are pretty much the same as in the preview icon. You can, however, give them longer names now. If you are using QL screen resolution, the names are truncated to 8 or 9 characters long, while longer names are shown on higher resolution screens. It is still not possible to load your own icons, you are still restricted to the admittedly wide range of icons built in. The author did say this was something others had requested and is looking at it for future versions, though it would increase

programming complexity. Since you can now use the Quick-launch menus to set up most programs, you will be putting less icons on the desktop than you might have done with the older releases, so this is less of an issue now anyway.

The utility menu has three new commands and an enhanced Pick facility.

The Clock command starts a digital clock going at the top of the Launchpad display. It's a simple digital clock like the toolkit 2 CLOCK command. There are some options to set USA or UK and Europe format, and whether you show only time or date for example. Interestingly, whatever you set is remembered for the next time you start Launchpad. Removing the clock is quite easy, just choose the command to disable it.

Command is for advanced users. Put quite simply, it sends commands to SuperBASIC or SBASIC. It seems to work by picking the BASIC interpreter and sending the command to the command line, optionally adding a linefeed to the command. For commonly used BASIC commands, this can save a lot of time manually selecting BASIC and typing in commands, although a lot of users probably won't need to use this command.

Lock is an unusual command. Quite simply, if the user concerned has a password to use Launchpad, it will lock everyone else out. Possibly useful in a workplace, to lock your system while you are away from your desk, or if you have young kids around who you wish to keep their hands off your QL while you answer the phone! Again, it's there for anyone who wants such a facility, my guess is it won't be used much.

Pick/Remove is an enhanced version of the old Pick menu. A list of programs running on the QL is shown, then you can either Pick or Remove that program by selecting the Pick or Rjob icons at the top then selecting the name of the program concerned. Sensibly, it defaults to Pick to prevent you accidentally stopping a program. Be very careful, I found to my cost it can even remove Launchpad itself! The list of programs includes all running programs, not just those started by Launchpad. Got to BASIC, EXEC a program in the traditional way, Launchpad lists that program in the menu. It can get into some difficulty if the program either has no name in the JOBS menu or changes its name since the menu list was displayed. It does have a refresh icon to update the program name list though. See Figure 3 for the Pick/Remove Jobs menu. Almost as an afterthought, a new menu called JOBS has been created which just brings up this menu, much quicker than going into the Utility menu and selecting it from there. In fact, switching tasks is such a common everyday function of Launchpad (QL is a multitasking machine after all) that I'm surprised it wasn't given its own icon at the top of the display rather than being tucked away on a menu.

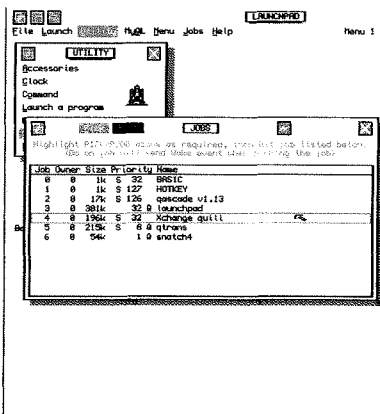


Figure 3 - The Pick/Remove Programs Menu

The File menu includes a couple of new commands. You can add a new icon to the desktop with the Add New Program command, and there is a new command to Make Directory on level 2 filing systems which support the MAKE\_DIR command. Current versions of Launchpad seem to have slightly changed the way in which you can right click on the desktop. Right click on what would have been a blank item in early versions would have let you install a new item, but now you can't right click on an empty space to do this, but some of the bugs in earlier versions seem to have been removed as a result of tightening up on this. You can still do things like the pseudo drag-and-drop of icons and so on, but the rules on what you can do seem to be a bit stricter now.

The manual is best described as 'long' and 'informative' - written in a matter of fact but helpful sort of way. It's divided into several parts, and there are separate manuals for all the accessory programs. I don't have strong feelings about the manual, apart from perhaps thinking that the sheer length of all the document files may put people off reading them. All the information is there.

There are some limitations such as the length of command lines you can send to programs. In most cases it won't matter too much, but might if you are sending complex command line parameters to C compilers and the like, perhaps.

You are still saddled with the same old colour scheme (white with bits of black, red and green) whatever mode you run it in. You simply don't get a choice of colour scheme, it wasn't feasible what with being written in compiled

Easypr basic.

Launchpad still makes the same demands on your system it pretty well eats memory needing over 300KB just for itself, huge by QL program standards. Pointer environment, toolkit 2 and plenty of memory are essential. It does change your way of working with QLs in many ways and you need to be prepared for that. That said, it allows you to go into BASIC and use your QL as you used to in many cases, yet provides the convenience and simplicity of a GUI interface. It's pretty smooth, well put together and consistent. It's very different from other GUI's I've used. There are touches of Windows and one or two other systems I've used, yet it's still quaintly unique and has that QL 'feel' about it - does everything you want and more yet still so easy to use.

In summary I'd say this is a solid product well worth the 20 pounds asked for it. It is not so radical that it jumps out at you in that "must have" sort of way that new technology often does, but has that good, solid, reliable feel to it, something that takes a little bit of getting used to yet remarkably simple to learn and use despite its size and the number of facilities offered. It's helped me change and modernise the way I look at my QL. There are one or two areas in which it's weak, such as the lack of colour schemes and the lack of ability to create and use new icons for program launching, but given how much the program has already improved so much during its fairly short lifetime, I'd hazard a guess even these might change in time! The author has a reputation for writing solid if unremarkable programs going back to the good old Page Designer programs and this is a very worthwhile addition to his stable of programs.

# Time To Renew

The next issue of QL Today will be the last issue of the current volume.

Volume 9 will start with the May/June issue, as usual.

Over the last 8 volumes we have had a lot of good cover disks/CDs etc. and we already have some things in mind for the upcoming year. We are dedicated to providing you with the most up date information and we try to keep the mix of articles in the magazine as varied and interesting as possible.

Keeping a good balance is something of an art and we can only achieve it if we have a lot of material to choose from. There have been issues where the content has been a bit too technical for some of our readers but we always try to make up for this in later ones. Pleasing all our readers all of the time is not an easy task.

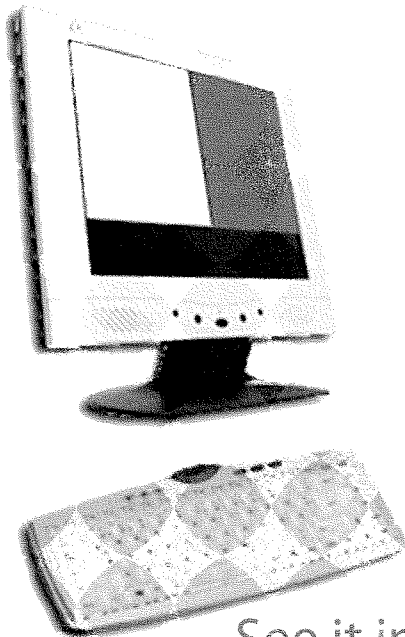
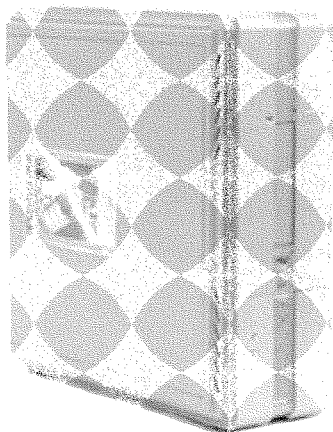
With your support, we will soon start volume 9 and ... wouldn't it be nice to be able to celebrate QL Today's 10th anniversary in 2005? Although prices for printing and postage have gone up again (as every year) we decided to keep the price stable. But we need a bit of help from you. Save us from wasting money by having to send out reminders etc.

We have added a renewal form. Please fill it in and return it a.s.a.p. If you choose to pay by credit card please use a card which does not expire before April, as we are not going to charge your card earlier than April.

Please also let us know if you do not wish to renew. We will be sorry to see anyone leave but, as a 'thank you' for the magazine in the past, please save us the expense of sending the reminder.

For German customers: Wenn die jährliche Verlängerung von QL Today automatisch vom Konto abgebucht wird brauchen Sie natürlich nicht antworten - es geht dann automatisch!

## NEW 2004 US-Specs Q60 A WHOLE LOTTA COOLER!



### Specifications

Silver/Blue Led ITX Case  
Electroluminescent Keyboard  
Flat LCD Screen w/ Sound\*  
2 IDE channels  
ReWritable CD  
Ethernet Card  
40 Gb Hard Disk\*\*  
Linux AND SMSQ/e\*\*\*

\* Optional

\*\* Size is optional

\*\*\* Can also include QDOS Classic with  
QubIDE 2 ROM if more than 1 HDD is chosen  
All Specifications are tentative - Watch QL Today  
for more information soon!

 **Quantum Leap**  
Hardware & Software

941 Lilac Street #1, Indiana, PA 15701. Email: ql.css@dokos-gr.net

See it in the US Show 2004

**Q60** Take the power  
back in your hands

# Hints and Tips for QPC

Jochen Merz

We have received various tips for QPC from a number of QPC users. Here is an extract of some very useful tips:

**BEEP Problems (1):** There have been two occasions when the beep in QPC has been quiet. On the first occasion I reinstalled QPC, checked the configuration was 99 and still no sound. It transpired that someone had played with the PC and turned the volume down to zero!

Recently the same silence fell over my beep. I first checked the volume and found it full on. Then I noticed that the Volume Control said it was being organised by Total Recorder. This had been added to my system to allow the recording of sound from the net. Setting its default sound system to the onboard Avance AC97 Audio the QPC beep burst back into life.

**BEEP Problems (2):** Previous versions of QPC do not generate a beep output anymore, although older versions did. There is no sound card in the PC, the system speaker was used to generate at least some kind of audio output on Windows 2000 and XP systems. BEEP is emulated properly in current QPC versions, but relies on a real soundcard to be available in your system.

**Display problems:** Under some circumstances on some systems, the QPC display turns black (especially if run in a window). As long as no windows error message is displayed, you should be able to get a proper display back by using CTRL F12 to switch to Full Screen mode and another CTRL F12 to switch back to windows mode.

**Floppy disk problems:** QL formatted floppy disks cannot be read under QPC. They cannot be formatted under QPC either. First, check that the disk is not write-protected! Obvious but it happens sometimes. DOS/Windows formatted floppy disks can be read and written to. So the floppy disk drive has to be working in general.

If nothing else helps, disable the virus scanner when using QL floppy disks. While doing this, do not insert floppy disks from other, maybe suspicious sources! As PC virus scanners are not aware of a "QL floppy disk format", some treat them as virus infected or unknown disk. The results are unpredictable, but a fairly common result seems to be in any kind of disk access (read, write, format) to be suppressed.

Do not forget to turn the virus scanner on after you stopped using QL floppy disks!

**Windows/DOS Floppy disks with long names:** the flexible floppy disk driver written by Tony Tebby auto-detects DOS and QDOS floppy disks and allows to read and write from and to both. In case of a DOS floppy disk, it restricts you to use filenames which are accepted by DOS (i.e. eight characters).

For example, flp1\_dirac\_filename.txt written to a DOS disk requires a subdirectory "dirac" to be present, in which the file "filename.txt" will be stored.

It will not allow you to save the file to "flp1\_directory\_filename.txt" (subdirectory is nine characters long) or "flp1\_verylongfilename.txt".

Windows (and more recent DOS versions "under" Windows), however, allows you to store long filenames and directory names. Under DOS, they appear abbreviated by a ~ (tilde) character.

The question is: can you write to a DOS disk using long filenames? The answer is: yes! ... configure one of the WIN drives to point to A: (the floppy disk) instead of a harddisk. You can do this in the QPC configuration dialog or during run-time, e.g.

```
DOS_DRIVE 8, "A:\"
```

If you have a DOS disk inserted, you can try  
DIR DOS8\_

and you should get a proper directory listing. You can now

```
SAVE DOS8_verylongname.txt
```

and do a DIR again. It worked!

Now try

```
DIR flp1_
```

(which will access the same floppy disk!) ... uh, the file is not there! Here comes the SMSQ cache system into play... so in order not to screw up the disk contents ALWAYS access the disk ONLY through one device, EITHER flp1\_ or dos8\_, but never mix them! You can get rid of the cache by typing

```
DEL_DEFB
```

but you have to make sure that you clear the cache before every read and write operation when you switch the device access.

However, if you do a

```
DIR flp1_
```

now, you should see something like  
verylo~1.txt

The obvious thing to try is

```
FORMAT dos8_
```

but this, unfortunately, report "not implemented".

Would be nice if this would work, Marcel!



# London Quanta Workshop, Jan. 2004

Roy Wood

The London Quanta workshop was held on a windswept rainy Sunday in January and that may have contributed to the scarcity of attendees. It was held in the Union Chapel in Southwark – the same venue as the previous year. This is quite a small hall under a church but it is a reasonably good venue for the workshop and, this time at least, we did not have to raise the slumbering Morris Dancers in order to be able to get the place set up.



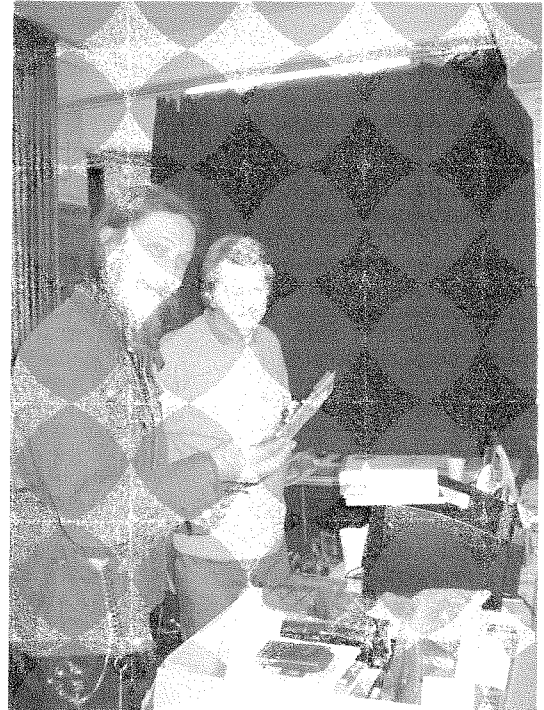
Some of the usual QL traders were unable to attend this time so Tony Firshman and I were the only two to have stalls there apart from the Quanta stand manned by John and Sara Gilpin who had struggled down from Manchester for the day. In spite of the small number of users in attendance there were some interesting discussion going on. Chris Cave was positively chomping at the bit (or byte) wanting to be able to use the new colour

interface for his programs but unable to find a way to access them from 'C'. He said that there was, apparently, a 'C' interface for this but it had not yet been released. I know that he will try it out when it does arrive so perhaps we can persuade him to write something for the magazine about it. At an early stage in the day we were asked to be quiet for a short

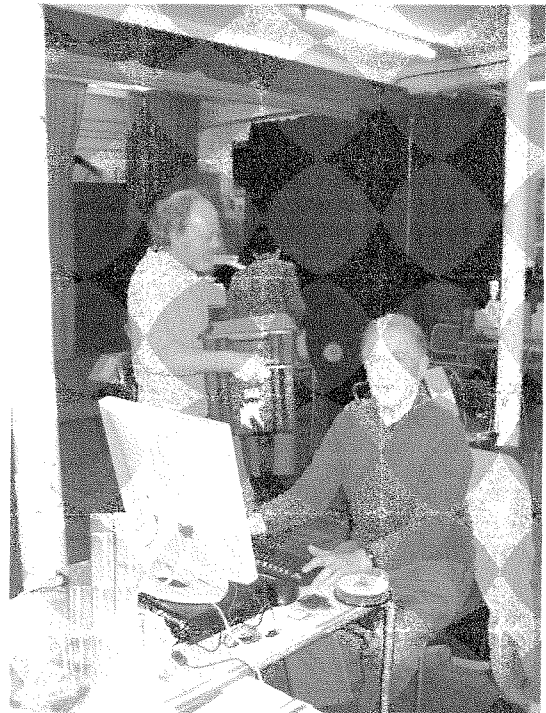
time so that they could hold a service in the church above. I was, at the time, demonstrating QCoCo to the assembled users (see *Byts of Wood*) and, having gone through the saga of crashes and bugs they were much

amused to see it crash on the first key press. (A problem that Wolfgang Uhlig has now solved. Tony, in particular, was laughing so loudly that he must have disturbed their Matins. Keith Mitchell was showing off his 17" TFT screen and micro computer running LINUX and UQLX although we both had

problems reading David Giliam's Q40 LINUX disk. There was a lot of interest in the Text 87 patch and in the demonstration of the SMSQ/E interface with its high colour icons and 3D buttons.



In general it was a pleasant day and deserved to have been attended by more users. I think that next years show should maybe be a bit later.



Christmas being well and truly over and the New Year (and the QL's 21st Birthday) on the horizon I find myself sitting at the keyboard pondering what I should be putting into this column. It is, after all, fairly traditional for there to be a retrospective in the first column of the new year. There are quite a few things to remark on and quite a few advances have been made this last year so I suppose we should really start there.

### No Ruby Con

The colour drivers have now left us balancing on a delicate point. This, to be brutally honest, is where the new generation of QL emulators and hardware waves farewell to the old QL. The old black slab is not going to run the new systems and software in the same way that the newer hardware and emulators do. I, for one, think that the new colour schemes and facilities are as much a Quantum Leap for our Operating System as the original QL was in its day but there are some users out there who still use the old QL as a main system. and they may be feeling rather left behind by the tide of innovation. I am often contacted by users who still have a standard QL with only a Trump Card or, at best, Gold Card and who still get a great deal of use from their systems. To be honest about it, if all you want to do is to write a few lines of text, maintain a small address database or any other of the more mundane aspects of compu-

ting, you can do this just as well on standard QL as on a fast, high powered PC. In some cases - given the complexity of modern software - better.

For those in the middle position of the QL hierarchy the situation is somewhat more fraught. Moving up to the high colour versions of SMSQ/E and running them in that mode will leave some software non-functional or working, but unable to display the results in a coherent or readable fashion. One of these was Text 87 which was my favourite word processor for many years. The patch released by Marcel Kilgus has changed that for QPC2 and high colour QXL users but the screen resolution on both the Q40/60 and the new Aurora colour drivers is different and so it will not work. In the case of the Aurora the screen driver is limited by the hardware to run in the mode it does and the reasons for the difference in the Q40/60 screen driver are mostly due to the way in which it was designed and developed. These were not some conscious effort to be different but also made hardware sense at the time as far as I can recall. Needless to say this is a somewhat confusing situation for the user to be in.

### A Patch on the Original

While I am on the subject of the Text 87 patch I would like to say that it is a simple and painless task patching the main program to work in High Colour mode and the different colour scheme on offer is very good. If there are Q40/Q60 and

Aurora users out there who would like to have the patch for their system they should contact Marcel who says he will undertake the task if there is enough interest.

### The Up Gradient

What most of the above means is that, if you upgrade your SMSQ/E to that latest version and you want it to run using the high colour mode, then you have to upgrade the other programs that you use as well. Many of the older programs will work perfectly well under the high colour mode SMSQ/E but some have display problems and need upgrading. One program that does need to be changed is Sysmon but that has been the case since Marcel changed the memory mapping.

The new colour schemes allow a much more 'modern' looking system and you will not get the full effect of the changes without upgrading the programs. Luckily most of these upgrades are pretty cheap so the overall cost to the user is not high and end result is a much better interface. Some programs are still awaiting changes but the ones I have in use are greatly improved by the new 'look'.

### Open Source Does the Business

A report out recently suggested that the business community is waking up to the realities of the Linux world, although the sources for the report were un-named and could, therefore be regarded as suspicious. The suggestion was that, while the companies saved money on the purchase of software, the eventual cost of re-training, productivity loss due to unfamiliar interfaces, data file incompati-

lity with other, more mainstream, software and other factors outweighed the saving. It states that the purchase price is only 10% of the 'Total Cost of Ownership' and the larger the company the bigger the downside of switchover. The reason I bring this up here is that another part of the report talks about adapting open source software.

The ability to modify software to your own purposes and fix bugs is an attractive proposition, especially when larger software houses are slow at responding to bug reports and ignore requests from day to day users. It does, of course, presuppose that you have the ability and knowledge to do this.

It goes on to mention that, once you have made these changes and submitted them to the ruling body of the software, your particular modifications may not be implemented and, perhaps, a conflicting piece of code may be incorporated in the next release rendering all of your work worthless, or even worse, destructive. If you persist with your own version, and there is nothing in the rules that says you cannot, you have then a 'fork' in the code. It somehow struck a chord with me regarding the debate on the open source nature of SMSQ/E and it was interesting to see that the commercial world probably spent a fortune on consultants to produce a report saying pretty much what we decided when the debate was raging. The chances of many 'forks' in our system are pretty low given the smaller user base but what is true of the macrocosm is also relevant to the microcosm.

I also received an email from

Duncan Neithercut (more on him later) who said he was very pleased with the way the SMSQ/E had developed since the licence. This made me think that, after the furore when the licence was being thrashed out I had heard no words of complaint and many of praise for the course the system had taken (OK so a few of those came from me but that is my purpose in writing these columns. I do try to say what I think and to keep this column's views as separate from the 'commercial' activities of QBranch as possible.) On the whole I would say that 2003 had shown that our decision to keep a strict licensing position and maintain the code in the way we do has borne a lot of fruit and I sincerely hope that 2004 will see even further development.

### Flash – Is He Going to Save the World?

While he was chatting Duncan told me he was working on some code to allow the Q40/Q60 to read FAT formatted Compact Flash disks. Now don't all jump at once because this is on the drawing board and may be a long way off being completed but he did say he had limited function already. This would be a good development since it would allow QPC2 users to write a QXL.WIN file to compact flash using one of the many adaptors available for their PC and then transport the contents onto their Q40/Q60 in one go. Since I recently rebuilt a Q40 system here I would have found that to be a very useful utility. The Compact Flash disk would then be a Super Floppy Disk

You may be surprised that I rebuilt the Q40 I had here but the reason is that formatting and

copying to floppy drives on the PC under QPC2 is so slow. This is not a fault of QPC2 but a result of the dreadful interface between the PC and the Floppy drives. Most of the physical components of the PC/Floppy I/O have not changed in 10 years and, as I mentioned before, many Floppy cables still have the old 5.25" Floppy connector on the them. I found I was wasting a lot of time at shows just writing the disks so I brought the Q40 back into play. I also needed to try out the new version of SMSQ/E before release.

### QL2004

Last year there was much talk of a large QL Workshop being held at some point this year to commemorate the 20th anniversary of the release of the QL. Many of us thought that this would be a good thing and there was a questionnaire inserted into the magazine to ask where people would like this to be held. Well we are now in 2004 (in case you had not noticed) and no venue or date has been announced. The indecision about the event has caused a few problems. I usually organise a workshop in Hove. This has, for the past five years, been held early in the year but the London show being held in January meant that I was forced to re-think the date and now I am not sure when the QL2004 show will be so I have no idea what date to aim at.

Geoff Wicks has been pushing for a decision on this for some time and was one of those who wanted the show to be held in Eindhoven so that the European users would be able to visit as well. On reflection I would say that many of the UK users are a bit older than their

continental counterparts and they would be unlikely to undertake a journey across the channel but Geoff has also proposed two shows being held. One in Eindhoven and one in the UK and this may be the best bet.

I would be very willing to attend both shows (provided the dates do not clash with other things) but we do need a decision soon to allow time to plan. And talking of planning ahead here is a new feature for this column.....

## Old Wood's Almanac (Predictions For This Coming Year)

**February:** Marcel Kilgus is found unconscious after a freak accident with a super fast version of QPC2. When he recovers consciousness he says he renounces all forms of computing, fast food and fizzy drinks. Goes to Tibet to train as a monk.

**March:** Dilwyn Jones, inspired by Arnold Schwarzenegger takes a body building course and gets elected President of Wales. Immediately declares all PCs illegal.

**April:** The EU declares Darren Brannagh's house an EU QL mountain and makes it an area of outstanding binary. QLs are declared an endangered species and no-one is allowed to throw one away.

**May:** Quanta receives a letter from Bill Gates saying that he is so impressed with the way the QL has lasted for 20 years that he is giving half the profits of his company to Quanta to use how they want. Robin Barker buys a stretch Limousine and takes the committee out for a

slap up meal before booking the Albert Hall for QL 2004

**June:** QL DVD walks away with multiple Oscars. Steve Rayel for 'Best Director in a film featuring QLs', Mplane for 'Best supporting role' and Bill Richardson for 'Best Newcomer'

**July:** The QL has a birthday party and wakes up the next morning with a hangover, in bed with an Atari ST. Its volatile memory is erased. Will it ever know if....

**August:** Tony Firshman enters the Tour de France riding a bicycle with a special 'Hermes' saddle designed to stop 'bum bounce' and give a faster throughput.

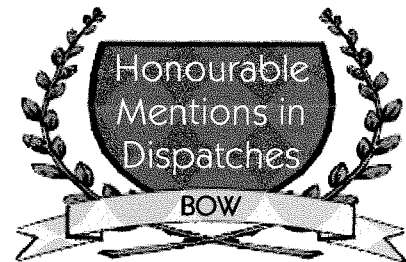
**September:** Quanta learns that the Bill Gates in question is William Eustache Gates and the company he owns is a wet fish emporium in Skelmersdale. Total profits for 2003 were £10 3s 6d in old money. Oops!

**October:** Chinese authorities, realising they manufacture 90% of the worlds motherboards and other computer components, change their calendar and declare this to be the 'Year of the Mouse'. They immediately come down with Repetitive Strain Injury and have to rest up for a while.

**November:** A particularly nasty computer virus wipes out 98% of the worlds computers and the internet stutters to a halt. Hordes of bleary eyes, white skinned gamers emerge, blinking into the daylight and spontaneously combust due to the exposure to the sun's rays. In desperation the world's governments fund Nasta's 'Super QL' project and, using Jonathan

Dent's TCP/IP and internet software rebuild the Internet better than it was before.

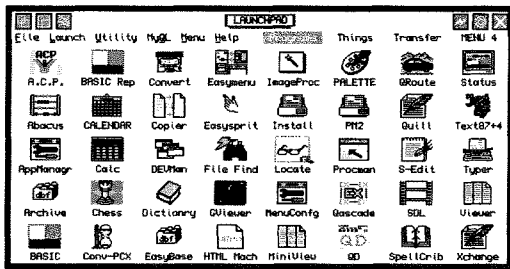
**December:** Marcel Kilgus is struck by Lightning whilst ringing the prayer bell in the Himalayas. He is rushed back to Germany unconscious and when he comes round he writes a 32 bit colour version of QPC2 for a Woolworths pocket calculator and goes out with the other traders for a meal during which he consumes a large Pizza and a bucket of Coke. Reclusive QL user dies and leaves Quanta £4.5 million. Robin is released from debtors prison. Normality is restored and the QL world has a joyful Christmas.



## Honourable Mentions in Despatches

The first accolade of the year is long overdue and goes to someone who had the task of writing a new colour configuration tool thrust upon him. Wolfgang Uhlig accepted this with a good grace and went on to produce what is, definitely, one of the best looking and most functional QL Tools I have seen. As the person who nominated him for this role I have been the recipient of most of the beta versions so I have watched it develop from a lot of 'Not implemented yet' messages to a fully functioning program. It uses windows and buttons to display the various items that can be changed and these then change to the colours selected by the user so, for

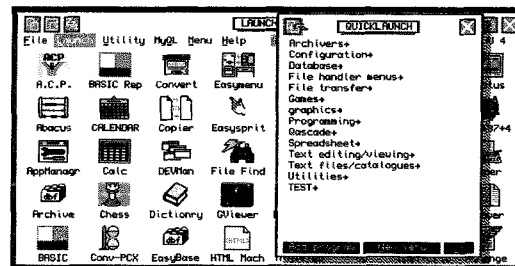
# THE LAUNCHPAD IS NOW READY FOR TAKE-OFF...



An all-new graphical user interface for QDOS or SMSQ/E from Dilwyn Jones. Fed up of typing in EXEC 'filename'? With this pointer driven application, simply set up desktop icons to launch programs, or create menus or "lists" of programs to launch, all in a simple to use no-nonsense graphical "point and click" system – use with mouse or keyboard. Runs on anything from expanded memory QL to QPC2 or Q60, as long as you have pointer environment and Toolkit 2

- Set up icons for launching your programs with a single mouse click from one of four desktop surfaces
- Set up your own program launching menu
- Up to 16 users, all with optional passwords
- MyQL menu to customise and remember your QL settings for each user
- Runs as a simple job, does not prevent you using BASIC or anything else
- Several accessory programs (calculator, calendar, screen saver, file handler, games and so on)
- Revolutionise your QL system – you'll wonder how you managed without it!

**£20.00**



## THE QL CD-ROMs ARE STILL AVAILABLE...

QL Emulators CD £5.00 - QL Documentation CD £5.00 - QL PD-CDR £5.00  
 QL PD Library CD £10.00 - QL Religion CD £10.00 - QL Literature CD £10.00  
 Line Design Clipart CD £10.00 - Famous Faces Clipart CD for Line Design £10.00  
 PD Software Library catalogue available on my website <http://homepages.tesco.net/dilwyn.jones/index.html>

## Want To Know More?

Visit the Launchpad web page at:  
<http://homepages.tesco.net/dilwyn.jones/launchpad/launchpad.html>  
 from where you can download a free trial version, limited only by the number of programs you can set up to run on it – more than enough to try out Launchpad.

Launchpad is available (price £20.00) from either the author:

**Dilwyn Jones**, 41 Bro Emrys, Tal-y-bont, Bangor, Gwynedd, LL57 3YT, U.K.

email: [dilwyn.jones@tesco.net](mailto:dilwyn.jones@tesco.net)

Payment in Pounds Sterling only

or from:

**Q-CELT COMPUTING,**

The Falconry, Glenmacnass, Glendalough, Co. Wicklow, Ireland

email: [darrenb@esatlink.com](mailto:darrenb@esatlink.com) (Payment in Pounds Sterling or Euros)

those items which do not need to be completely refreshed the result is instant. Many of the ideas for this are quite innovative and are better implemented on the QL than on other systems I have seen. Similar programs under Windoze allow the user to select a colour from a swathe of different colours but Wolfgang's program, leading on from his original 'Colour Tools' of a couple of years ago, actually allows you to modify the end result in three different areas, hue, brightness and saturation. Using these three criteria you can create any number of different shades and colours. These can then be stored in a 'My Favourite Colours' Bar which allows you to re-use the colour on another item. The Micr\$oft version does not allow this and you have to re-select the colour each time. Bonus for the QL ingenuity.

The results of your labours can be saved as a 'Theme' which can then be loaded in the boot file or at any other time during the computer's use. There is a button which will allow you to make the theme you are working on the current one so you can see the effects on the programs you use. Another neat idea is the inclusion of a 'Synchronise' button which will offer to choose colours to fit those which you have just changed. This makes a coherent 'Theme' easier to produce. All in all I have been very impressed with the way the program looks and the large amount of work he has obviously put into it. I hope that he will find the time to produce some more programs for us.

Oh, and by the way, it is free!

You can download it from:  
<http://www.uhlich.nl/ql/index.htm>

I can feel a competition coming on - best 'Theme' anyone?

Wolfgang deserves a lot of praise for the innovative way he has tackled the task and a massive 'Thank you' from the SMSQ/E community for his work. All together now.....

## A Bugs Life

As an aside to this I would like to put myself forward for 'Big slap on the wrist - must be more methodical' prize.

Wolfgang sent me several versions of this program and, as I said they all got better with each new version. Shortly before the London show I got a 'Final Release version'. This seemed to work well and I started to put together a disk with all of the colour stuff on it. Then I found a bug. I clicked on one of the menus chose to change the colour and, bang! Red Qlib screen! Okay I contacted Wolfgang. 'No works here' was the reply. We decided it had to be something in my BOOT file being loaded which stopped it working. Fire it up with a small BOOT file. It works. OK must be an extension. I spent an entire evening deleting lines from the BOOT. Every now and then I got it to work and sent an excited email to Wolfgang saying I had found the problem only to find that the next time it did not work. Finally I worked it out. The program has two distinct sets of buttons, one on the right and one on the left. The left hand ones work on the big items like Application Windows and Menus. The ones on the right work on smaller items like

Scroll Bars Buttons etc. They both call the same colour configuration windows but the routines that call them and return the values are different. The ones on the left worked and the ones on the right didn't.

It was that simple. I had just fired it up and chosen an item at random thinking that the bug was in the colour selection. As soon as I had told Wolfgang about this he reproduced the bug and solved it - and promptly introduced another one. This was, by this time, Saturday afternoon before the London show where I wanted to give away disks and demonstrate the program. Frantic emails passed between us until we finally had a working version of the program - all bugs expunged.

While my machine booted up I told Tony Firshman, Keith Mitchell and other assorted people at the show the story and said you will see something brilliant! I fired up the program and said 'When I did this last night it crashed but Wolfgang has fixed it and....' bang it crashed! Tony was apoplectic with laughter. I had managed to find the one remaining bit of code that had not been fixed with the first click and in front of an audience. Serves me right eh?

## One Last Thought

Someone told me a joke recently and I thought it would work well adapted for the QL traders and programmers:

**Q:** How do you become a Millionaire dealing in QL software?

**A:** Start off as a Billionaire.

**Have a Happy and Preposterous New Year!**

# QL 2004

Geoff Wicks

Last year's QL Today poll on the venue for QL2004 proved to be almost as close and controversial as the last American presidential election, but without hanging and pregnant chads to reduce the tension. The QL community was divided with the continentals wanting the show in Eindhoven and the UK-ers wanting it in the UK. It made grim reading for an organiser, because, whatever he did, he was likely to upset half of the QL community.

After the poll an eerie silence descended on plans for QL2004. I had hoped that at the Byfleet show there would be some information, but instead there was confusion. People who had attended the show told different stories about QL2004 plans.

I am a QL2004 maverick. Although I live in the UK, I firmly believe the continent should have the show. The UK ran QL2000 and now it is the continent's turn. Also the centre of QL development has moved from the UK onto the continent and, in particular, Germany.

On New Year's Eve I asked for reaction from the QL-users email group about a possible Eindhoven QL2004 show. I had just 2 replies. One, sent in the first hour of the New Year, was from QUANTA. I took this to mean that should there be a clash between QL2004 (UK) and QL2004 (Eindhoven) negotiation was possible.

There was a green light for QL2004 (Eindhoven), but please stay reading UK QL-ers. You are not forgotten.

Eindhoven is an ideal setting for a continental QL2004, because it is readily accessible from the Netherlands, Belgium and North Germany. It is also within easy

travelling distance of France, South Germany, Austria, Italy, Switzerland and the UK. In the past Eindhoven was well known for its international QL shows at the Pleincollege St. Joris, a secondary school with ample facilities for a large group of people. The local user group, Sin-QL-Air, obviously need the time to discuss the possibility of hosting QL2004 especially as they are now a fairly small group. We have also given them short notice of our wish to hold the show in Eindhoven. Obviously there are many practical problems we have to discuss with Sin-QL-Air but if QL2004 goes ahead it will be held on a Saturday in October or November. Sin-QL-Air have already suggested a possible date, but this date was not possible for Jochen.

QL2000 was a luxury show that is reputed to have cost QUANTA over £2,000, but we have to organise QL2004 without financial backing, which means that we have to be realistic about the possibilities and limitations. Fortunately the location will cost us nothing. Refreshments will have to be self-financing, and how lavish these can be is still to be negotiated. There will be no organised program for spouses and other guests, but the Netherlands is full of historic towns within easy reach of Eindhoven. We have, as yet, no plans for an official show dinner, but there is a long after-show tradition of a visit to a Chinese restaurant.

What can you expect of QL2004? We want this to be a nostalgia show to which we invite ex QL friends for a reunion. This theme was inspired by a group of Scandinavians who replied to my New Year's Eve email. They are no longer active QL-ers, but they wanted an opportunity to meet old QL friends. We want to show ex-QL colleagues how far the QL

has advanced in the last few years. This means plenty of Q60s and QPCs running programs in the new colours, but also, Wolfgang Lenerz has suggested, some black boxes to remind people of how we started. Maybe existing QL-ers also need a morale boost to remind us we are a viable force with plenty to boast about.

Initial reactions from QL-ers indicate they want a program of lectures and demonstrations. Jon Dent has suggested an Internet connection workshop dealing with registering with an ISP and emailing from your QL - a good follow up to the theme on the last day of QL2000. If it is technically possible we could consider some form of internet linkup with QL-ers around the world.

I would like QL2004 to become a "Peoples' Festival" with a demonstration program that could change as the day progresses, but we also need items to form a basis program. Any suggestions for these would be welcome. We also need you to contact your old QL friends and bring them along.

There are still many uncertainties over QL2004 and logistical problems to be overcome, but I hope many of these will be solved by the time you are reading this, and that we can give more concrete details in the next QL Today.

And UK QL-ers? If you also wish to have a QL2004 event you could organise a similar themed event without too many clashes with Eindhoven, but I have another suggestion. Why not think about QL2005? What about a huge QL 21st Birthday celebration in the weekend 15/16th January 2005? Perhaps a social event rather than a traditional workshop? And don't forget the UK birthday tradition. Who will provide the birthday cake and which celebrity will blow out the 21 candles?



# The QL Show Agenda



## AGM & Quanta Workshop - Manchester (UK)

**Saturday, 17th of April, from 1:00 pm and  
Sunday, 18th of April, 9:30am to 3:00pm**

**NEMQLUG organises the show  
at the traditional Manchester venue:  
3rd Davyhulme Scout Headquarters  
Conway Road, Davyhulme, Manchester**

**There will be a celebratory 21st dinner on Saturday night, deposit of  
£10.00 required by 15th March 2004.**

**Talks have been arranged on the latest hardware/software developments**

**Hotels: Coach House Travel Inn,  
Trafford Centre, Tel. 0161 747 8850  
Mersey Farm Travel Inn, Carrington, Tel. 08710 977179**

**The Scout HQ. is located close to Junctions 9 and 10 off the M60,  
formerly Junctions 4 and 3 off the M63.**

## US QL Show 2004, Orlando, Florida (USA)

**Sat/Sun., 24/25th of April**

**Quanta and NASQLUG are pleased to announce the US QL show to be held 24/25 April 2004  
Saturday 10 AM to 5 PM and Sunday 10 AM to 3 PM  
at the Days Inn International Drive, 7200 International Dr Orlando, FL 32819.**

**Phone: 407-351-1200, Toll-Free in US: 800-224-5057, Fax: 407-363-1182.**

**Their room rate is \$49.95 per night for 1 to 4 persons (contact Diane at the hotel directly for this price). (Plus 11.5% tax). However, rates as low as \$35.95 are available at Discount Hotels and Resorts US phone (800) 479-1406 or [www.discounthotels.cc](http://www.discounthotels.cc) Make sure to specify Days Inn at 7200 International Drive since there is another Days Inn on International Drive.**

**For more details see last page of previous QL Today issue or visit the website dedicated to the USA show by Jim Hunkins:**

**<http://www.jdh-stech.com/ql-usashow.htm>**